Improving Support of Conversations by Enhancing Mobile Computer Input

A Dissertation
Presented to
The Academic Faculty

by

Kent Lyons

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

College of Computing
Georgia Institute of Technology
August 2005

Improving Support of Conversations by Enhancing Mobile Computer Input

Approved by:

Dr. Thad Starner, Advisor
College of Computing
Georgia Institute of Technology

Dr. Jeffrey S. Pierce
College of Computing
Georgia Institute of Technology

Dr. Gregory D. Abowd
College of Computing
Georgia Institute of Technology

Dr. Bradley J. Rhodes
Ricoh Innovations

Dr. Elizabeth D. Mynatt
College of Computing
Georgia Institute of Technology

Date Approved: June 27th, 2005

ACKNOWLEDGEMENTS

This work is funded in part by National Science Foundation Career Grant #0093291, the DARPA "Augmenting a Program Manager" seedling, and the Wireless Rehabilitation Engineering Research Center on Mobile Wireless Technologies for Persons with Disabilities, which is funded by the National Institute on Disability and Rehabilitation Research of the U.S. Department of Education under grant number #H133E010804.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Mobile computing is becoming one of the most widely adopted technologies. There are 1.3 billion mobile phone subscribers worldwide, and the current generation of phones offers substantial computing ability. Furthermore, mobile devices are increasingly becoming integrated into everyday life. With the huge popularity in mobile computing, it is critical that we examine the human-computer interaction issues for these devices and explicitly explore supporting everyday activities. In particular, one very common and important activity of daily life I am interested in supporting is conversation. Depending on job type, office works can spend up to 85% of their time in interpersonal communication.

In this work, I present two methods that improve a user's ability to enter information into a mobile computer in conversational situations. First I examine the Twiddler, a keyboard that has been adopted by the wearable computing community. The Twiddler is a mobile one-handed chording keyboard with a keypad similar to a mobile phone. The second input method is dual-purpose speech, a technique designed to leverage a user's conversational speech. A dual-purpose speech interaction is one where speech serves two roles; it is socially appropriate and meaningful in the context of a human-to-human conversation and provides useful input to a computer. A dual-purpose speech application listens to one side of a conversation and provides beneficial services to the user. Together these input methods provide a user the ability to enter information while engaged in conversation in a mobile setting.

CHAPTER 1

INTRODUCTION

Mobile computing is becoming one of the most widely adopted computing technologies. Personal digital assistants (PDAs), mobile MP3 players, and digital cameras are increasingly becoming incorporated into everyday life. The next generation of smart phones offers substantial computing ability and is blurring the line between different mobile devices. Mobile phone text messaging already displaced two–way pagers. Cameras of increasing resolution are being integrated into phones, as are applications that were previously only on PDAs. Phones are also starting to incorporate removable storage devices, and manufacturers are building models that also serve as MP3 players.

Mobile phones are ubiquitous; there were 1.3 billion subscribers in 2004, and there could be as many as 2 billion by 2007 [3]. Wireless text messaging is widespread with predictions of a rate of over 1 trillion messages per year being reached shortly [33, 42]. These statistics are remarkable considering the inefficiencies and poor design of current text entry methods for mobile devices.

With the huge popularity of mobile computing it is critical that we examine the human–computer interaction issues for these devices and explicitly explore supporting everyday activities. In this dissertation we begin this process by examining the issue of mobile input in the context of supporting face–to–face conversations.

1.1 Mobile Computing

While mobile technology is becoming common place, it often falls short of its users' expectations and needs. Perry *et al.* [45] examined mobile workers' use of documents and mobile technology for the management of information. They found that laptops might be carried to a remote site but not from meeting to meeting in one location. They noted that "the physical form of these objects does not facilitate 'casual' carrying and prevents them from

being ubiquitously available to the mobile worker." While portable, the laptop does not facilitate true mobile use.

Kidd also explored characteristics of knowledge workers and revealed some of their information practices and hints at some potential limitations of mobile technology. She notes that the users of mobile electronic notebooks might not be comfortable with these devices for note taking even though information is critical to their work. Instead she speculates that they might be used for "non–primary aspects for their work such as noting a telephone number, a diary date or a short message for a colleague" [27]. This list maps well onto many of the familiar applications on personal digital assistants. There are several applications designed to store a person's schedule, reminders of important tasks, and other information useful for carrying out the activities of daily life.

Unfortunately, many users are often unsuccessful in using the technology for these tasks. Several studies have shown that people tend to revert to writing on scrap paper, post–it notes, their hand, etc. for "micronotes" and other short pieces of "notable" information such as phone numbers, names, and to–do items [32, 9, 21, 5]. Similar patterns in the use and disuse of mobile devices have also been observed for calendaring [58].

1.2 Wearable Computers

In contrast to the above work, many wearable computer users report that they do use their machines in the above situations where other mobile devices have been shown to fail. Anecdotally, these users report that they often take notes or retrieve information in a large variety of everyday situations. While wearables are still novel, a few researchers and hobbyists have adopted them into their lives. The use of this new technology is worth examining because it offers a unique perspective on mobile technology.

A wearable computer is a computer designed to be worn on the body instead of carried. Ideally, the computer is always with the user and often becomes highly personal. Computationally, they often offer more resources than commercial mobile devices and tend to be equipped with unique peripherals. Users are often seen in a wide variety of situations wearing their head–up displays (Figure 1) or typing with one–handed keyboards such as

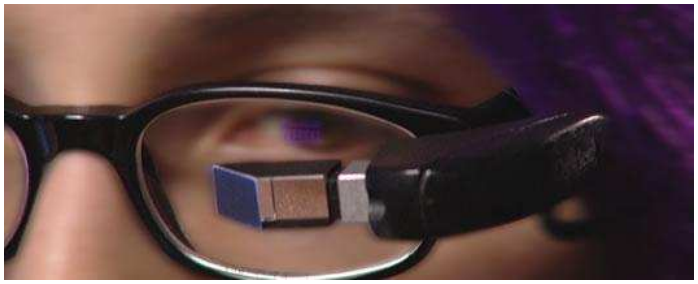

Figure 1: The MicroOptical head–up display mounted on eyeglasses.

Figure 2: The Twiddler one–handed chording keyboard

the Twiddler (Figure 2) [35, 57].

Prior to this work, we did not have a firm understanding of exactly what tasks the computers are supporting nor in what situations. Likewise, we knew little about how these users employ the wearable to accomplish those tasks. In Chapter 2 we present a case study of an expert wearable user to examine these issues. We found that the expert in our study used his wearable in a large variety of everyday situations as an information repository. Additionally, the wearable was used mainly in a support role and the user's attention remained focused on the real world. Finally, a majority of the interactions occurred while

the user was engaged in a social activity, usually a conversation. Given the extensive use of the wearable in social situations, we decided to investigate conversations more thoroughly.

1.3 Conversations

More broadly, research has shown that face–to–face conversations are important and common in everyday work. A survey of studies of office work found that communication dominates much of the workday [44]. In it, Panko found that between 25–85% of an office worker's time is spent in interpersonal communication. Managers spend 55–60% of their time in face–to-face or oral communication while this increases to 70–75% of the workday for CEOs.

Whittaker *et al.* found similar patterns in a study of informal workplace communication. In this study, they also explicitly examined where the conversations occurred. They observed that for their two participants conversations frequently occured in mobile settings; 17% of their participants' total work day was spent in conversations while "roaming" or away from the desk [62, 18]. Given the amount of time people spend in conversation that frequently occur in mobile situations, mobile computing appears to be a good platform to leverage in support of those conversations.

1.4 Mobile Input

The focus of this work is to enhance mobile computer input to better enable the support of conversations. While fully supporting conversation will require applications and interaction techniques tailored to the tasks encountered while in conversation, fundamental to any interaction is the ability to enter information into the computer. In this dissertation, we present two methods that improve a user's ability to enter information into a mobile computer in conversational situations. First we examine the Twiddler, a keyboard that has been adopted by the wearable computing community. Our second input method uses dual–purpose speech, a technique designed to leverage a user's conversational speech.

1.4.1 The Twiddler Keyboard

The first input method we examine is the Twiddler, a mobile one–handed chording keyboard made by HandyKey (Figure 2). This device has been adopted by many wearable computer users and offers rapid text entry rates. Wearable users employ it during conversations and use it as their input device while taking notes on points of interest and entering other commands to the computer (Chapter 2).

The Twiddler employs the same button layout as a mobile phone with a grid of three columns and four rows. Unlike a mobile phone, each row of keys is operated by one of the user's four fingers. Additionally, the Twiddler has several modifier buttons such as "Alt" on the top operated by the user's thumb. Users hold the device in the palm of their hand like a cup with the keys facing away from their bodies. All five fingers on a hand can be used to type. Unlike many other keyboards, the Twiddler is a chording keyboard. Instead of pressing keys in sequence to produce a character, multiple keys are pressed simultaneously. Each letter of the alphabet can be typed on the Twiddler by pressing one or two keys concurrently.

1.4.2 Dual–Purpose Speech

Our second input method uses dual–purpose speech, a technique designed to leverage a user's conversational speech. A dual–purpose speech interaction is one where speech serves two roles. It is socially appropriate and meaningful in the context of a human–to–human conversation and provides useful input to a computer. A dual–purpose speech application listens to the user's speech and provides beneficial services. Dual–purpose speech is also sensitive to privacy concerns. Our applications use a noise cancelling microphone which only picks up the user's speech and are designed to rely only on the user's side of the conversation.

1.4.3 Complementary Techniques

These two input techniques explore different points in the mobile input design space. First, they offer different benefits for novice and expert users. The Twiddler, much like any

keyboard, requires practice to achieve rapid text entry. Once the user learns to type, she can rapidly enter large amounts of text and in a large variety of situations. While we can facilitate learning, it takes practice for the user to become proficient.

In contrast, dual–purpose speech is more amenable for novice users. Early research by Gould found that a speech interface can be faster and more "natural" than typing or writing for dictation tasks [19]. He also found that speech systems may be useful for novice users. Speech may seem more intuitive and help occasional users of a system obtain the desired functionality at a speed faster than if they had to learn an alternative device.

While potentially useful for novices, the application of speech in a mobile device must be considered carefully. In particular if the computer is to be used in social situations, it may become socially awkward if the user must vocalize to the computer to provide input. With dual–purpose speech, we limit the use of speech to situations where a single utterance fits into the context of the conversation and also provides meaningful input to the application.

The second way these two interaction techniques are complementary is with respect to different turns of the conversation. From our experience with the Twiddler, we have found that users primarily type and take notes while listening to their conversational partner and pause their typing while speaking. Similar behavior occurs with more traditional technology such as pen and paper; a user can write notes while listening but usually pauses when it is her turn to speak. This behavior occurs because it is difficult to type (or write) and speak at the exact same time because this would require two verbal productions [51]. Instead, a user must serialize her activity and type just before or after speaking.

In contrast, dual-purpose speech is designed to be used while speaking. Unlike the Twiddler, the user simultaneously communicates with a conversational partner and computer using dual–purpose speech with a single verbal production. Furthermore, it would be inappropriate to use this technique while listening to someone else because speaking would be socially disruptive.

1.5 Thesis and Contributions

The preceding discussion leads to our thesis statement. After introducing our thesis we enumerate our contributions and discuss how each contribution is supported by a chapter of this dissertation.

1.5.1 Thesis Statement

Our hypothesis is that we can enhance mobile input during conversation via two complementary methods:

- By increasing a user's data entry capability with the Twiddler chording keyboard, and
- Through reusing conversational information with dual–purpose speech.

1.5.2 Contributions

We explore this thesis through our contributions which include

1. A case study of an expert wearable computer user and an examination of the use of the computer in everyday situations (Chapter 2).
2. Research determining the learning rate of the Twiddler and a comparison to the common mobile phone entry method of multi–tap (Chapter 3).
3. An examination of expert characteristics of Twiddler chording including research on the effects of multi–character chords (MCCs) and limited visual feedback (Chapter 4).
4. An evaluation of improving novice use of the Twiddler through use of a chording tutorial (Chapter 5).
5. The input technique of dual–purpose speech and three example applications: the Calendar Navigator Agent (CNA), DialogTabs, and Speech Courier(Chapter 6).
6. An evaluation of a dual–purpose speech application (the CNA) designed to uncover the relative tradeoffs between the use of traditional pen input and speech input in the context of a scheduling conversation (Chapter 7).

1.5.3 Overview of Dissertation

First, in Chapter 2 we discuss our case study of an expert wearable user designed to reveal how an expert user employs his machine in daily life. We examine the use of the computer by collecting periodic screen shots of the wearable's display and utilize these screen shots in interview sessions to create a retrospective account of the machine's use and the user's context. This study reveals several key points. First, the wearable is used in a large variety of situations as an information repository. Second, the wearable is used mainly in a support role. The user's attention remains focused on the real world. Finally, a majority of the interactions occur while the user is engaged in a social activity, usually a conversation.

Next, we explore increasing a mobile computer user's data entry capability with the Twiddler keyboard in Chapters 3, 4, and 5. We chose to explore the Twiddler in detail because the expert from our case study demonstrated that he could successfully use the Twiddler in conversational situations and can quickly take notes on points of interest.

In Chapter 3 we present a longitudinal study of novice users' learning rates on the Twiddler. Ten participants typed for 20 sessions using two different methods. The first method is to use one–handed chording on the Twiddler while the second is multi–tap, a common mobile phone text entry method. We found that users initially have a faster average typing rate with multi–tap; however, after four sessions the difference becomes negligible, and by the eighth session participants type faster with chording on the Twiddler. Furthermore, after 20 sessions typing rates for the Twiddler continue to increase.

We continue our longitudinal study of the learning rates for chording on the Twiddler in Chapter 4. Five of our original ten participants continued and achieved an average rate of 47 wpm after approximately 25 hours of practice in varying conditions. One subject achieved a rate of 67 wpm, a rate equivalent to that of the expert from our case study in Chapter 2. We also analyze the effects of learning on various aspects of chording, provide evidence that lack of visual feedback does not hinder expert typing speed and examine the potential use of multi–character chords (MCCs) to further increase text entry speed.

In Chapter 5 we examine how two different techniques might be incorporated into a typing tutor to help improve a novice user's experience with the Twiddler. Specifically, we

examine the effects of a phrase set designed for the Twiddler and the manipulation of an on–screen keymap representation. Sixty participants were divided across 6 conditions and typed for two 20 minute sessions. We found that the ordered phrase set aids novice Twiddler typists' typing rate, error rate and mental workload while in use. Likewise, highlighting our on–screen representation helps typing speed, accuracy, and reduces workload.

The second part of this dissertation explores ways to reuse conversational material with speech recognition. In chapter 6 we introduce the concept of dual–purpose speech: speech that is natural in the context of a conversation while providing meaningful input to a computer. We motivate the use of dual–purpose speech and present three applications that assist a user in conversational tasks: the Calendar Navigator Agent, DialogTabs, and Speech Courier. All three of our applications are built so they can be used while mobile, since many conversations occur while roaming [62]. The Calendar Navigator Agent navigates a user's calendar based on socially appropriate speech used while scheduling appointments. DialogTabs allows a user to postpone cognitive processing of conversational material by providing short–term capture of transient information. Finally, Speech Courier allows asynchronous delivery of relevant conversational information to a third party. These dual–purpose speech applications reduce the amount of manual input and instead reuse material from the conversation.

Chapter 7 presents an experiment evaluating the effectiveness of dual–purpose speech in the context of one of our applications, the Calendar Navigator Agent. We examine the ability of novice users to enter information into a calendar on a personal digital assistant using dual–purpose speech while engaged in a scheduling dialog with a researcher. Twenty participants scheduled a sequence of appointments using two input conditions, speech and traditional pen input. We found that our speech condition did not show a performance benefit, but instead resulted in a conversation where the participant held the conversational floor longer. Overall, novice users quickly accommodate to the technique, and it offers an additional input option which can be employed while a user is engaged in a conversation.

Finally, we conclude with a discussion of how this research fulfils the claims of the thesis statement and present some possible directions for future work (Chapter 8).

CHAPTER 2

WEARABLE COMPUTER USE: AN EXPERT CASE STUDY

As discussed in the introduction, mobile technology tends to fail in everyday situations such as jotting down short notes from a conversation. In contrast, we have anecdotal evidence of successful usage for similar tasks with wearable computer users. In this chapter, we present a formative study designed to uncover the practices of a wearable computer user and explore how an early adopter takes advantage of wearable computing technology in everyday situations. In particular, we are interested in the situations in which an expert wearable user can employ his computer and the mechanisms that might contribute to his success.

For this study we collected data during the course of an expert wearable computer user's normal daily activities over a five week period. The data consists of periodic screen shots from the wearable's display and interview sessions which were used to create a retrospective account of the machine's use and the user's context. Using this data we detail the technology our participant employs as well as general characteristics of the computer's usage. We also present several examples of interactions with the wearable computer in everyday situations. Using these examples, we discuss trends in the data showing how the computer is used to augment the user's memory and in support of social or conversational tasks.

2.1 Method

We developed a method to accommodate the everyday nature of our participant's wearable computer use. Our data consists of screen shots captured on the wearable computer augmented with interviews of the participant. We chose this method because it is difficult to directly observe the interaction between the machine and user because the head–up display removes an observer ability to see what the user sees. Capture on the wearable enables us

to gather information directly about the interaction with the computer [34]. Furthermore, the user operates in a wide range of environments making representative direct observation of wearable use logistically impractical.

Our participant's existing wearable computer was augmented to capture screen shots of the information presented on the user's head–up display to the hard drive approximately every five seconds. After some initial experimentation, five seconds was chosen to minimize the impact on the user's machine while still maintaining a high rate of capture. The screen shots were later used in interview sessions to create a retrospective account of the machine's use [4]. During these sessions, we played back the captured log to the user as a movie while often stopping and revisiting portions of an interaction. The participant detailed how the machine was being used, and the interviewer asked questions about general context such as who was around, his location, and the current activities. We believe the screen shots played an invaluable role during the interview sessions because everyday tasks become tacit. The screen shots served as a cue to remind the user of what he was doing instead of attempting to recall what happened. The recorded log provided an objective record of what happened and how often.

Because the wearable is a very personal device, potential access of private information in the course of daily use such as passwords, sensitive email, and medical records is an issue. Our solution to censoring this private information was to give the user the ability to control the capture software. The user could pause the logging if he was working on private information for an extended period of time. Additionally, the user could also black out screen shots already logged if he realized sensitive information was recorded.

Sample size is an issue when studying wearable computer use especially given our particular interest in everyday use. There are only a handful of people in the world who have adopted wearables into their lives and have continued to use the computers daily. Obviously, with such a small user population one could not hope to span the possible space of wearable computer use. The small number of current users may not be representative; however, they are using their computers while doing typical everyday tasks working with information and managing their daily lives. As a result, instead of attempting to generalize

across a very small number of users, we are seeking to understand the practices developed by a single successful wearable computer user. This work lays the foundation and provides motivation for the research presented in the following chapters.

2.2 Case Study Data

Our participant is in an academic research environment where he uses the machine routinely in a large variety of situations and has been doing so for over eight years. During the course of our five week study, we collected 68 hours of use with the machine from 15 different days. This is approximately 15,000 screen shots. The wearable was used in a large variety of situations, and after exploring the data, it became clear that the situation influenced how the machine was used. In addition to being used while alone, the wearable was often used while engaged with other people. This could be in the form of one–on–one meetings, small groups, talks, demos, or impromptu gatherings.

Most of the usage of the machine occurred in the user's academic work setting. The machine was used in the user's office, the hallway, the social area near his office, the lab, and conference rooms. The user also spent some time working in another building across campus using the machine in classrooms and around the building. During our study, the user also went on two trips to visit other research institutions, one in a foreign country. The machine was used to prepare for these trips and for support during the trip. In the interviews, the user also indicated that he used the machine while riding on a train, as a passenger in a car, and while walking. The wearable contains a wide variety of information including notes, email, to do lists, contact information, and personal records. It was also used as a scratch pad, and on a few occasions for writing and editing articles.

2.3 Expert's Wearable Computer

At the time of our study, our participant had been using a wearable computer daily for over eight years. The computer is a derivative of the Lizzy design [57] and is housed in a bag worn over the shoulder and rests on the user's left side by his hip. This arrangement allows the user to continually wear the machine throughout the day. The Twiddler2, a one handed

chording keyboard, is the input device (Figure 2). It serves as a combination of keyboard and mouse; however, the participant only utilized the keyboard functionality during our study. The display is a MicroOptical CO-3 VGA head–up display with 640x480 resolution and is designed to mount on a pair of eyeglasses (Figure 1). The user modified the mount so he could quickly attach and remove the display as needed. Finally, the wearable is designed for low power consumption so that it can be powered throughout the day. The user reports the computer typically runs ten to twelve hours on a set of batteries, and he swaps out batteries as needed for a longer runtime. Together, these design features allow the user to call the machine to action quickly at any time by snapping the display to the user's glasses and grabbing the Twiddler from his side.

Our participant's wearable computer runs Linux and the X Windowing System. Emacs is the primary application used, and the vast majority of interaction with the machine happens within this versatile text editor. For the few occasions where the user did not directly interact through Emacs, an xterm was opened and used temporarily. This occurred when the built–in Emacs shell was not sufficient at displaying the needed application. It is interesting to note that the user did not run any software explicitly designed for wearable use during this study.

Figures 3 through 6 show typical screen shots of the user interacting with the machine.[1] Emacs fills most of the user's display. Xclock runs in the bottom right corner of the screen but is partially covered by the Emacs window. As a result, only half of the clock is visible. The user indicated that when he recently changed the font for Emacs it covered up the clock, and he had not yet fixed it.

Within Emacs, the line of text at the bottom of the screen in inverse video is the mode line. This line shows various status information such as attributes about the current state of the file (modified, saved or read–only), the name of the file being edited, the time, and the CPU load of the machine. In parentheses, information about the current mode is displayed. The last two items show the current line number and the percentage from the top of the file.

[1]The figures have been altered for anonymity and readability.

2.4 Usage Examples

We next introduce the data collected through examples. Some of these are common activities for this user, while others are more rare but demonstrate the range and richness of the situations in which the wearable is used. These categories of examples emerged upon analysis of the data. They reflect the patterns in his information and in turn the patterns of wearable use in different everyday situations.

2.4.1 "Today"

Our first example of a typical interaction with the wearable centers on the "today" file. This file acts as a very flexible to do list that is instantiated as a free form text file and contains short–term important activities that have little meaning long term. This file is not intended for archiving; items are deleted as they are completed or become irrelevant.

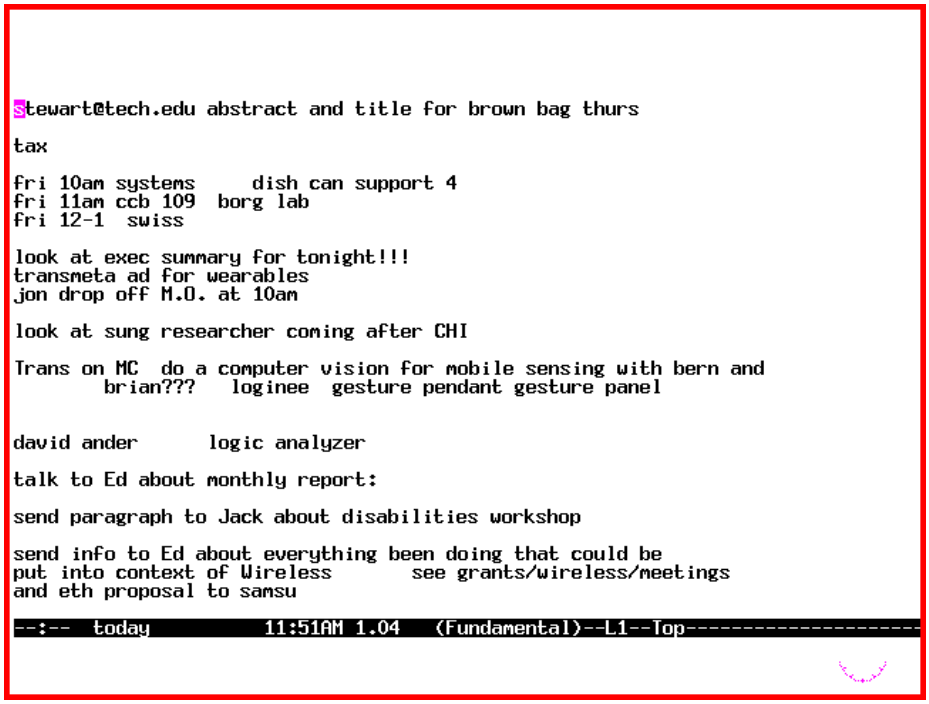

Figure 3: The "today" file contains brief notes on to do items.

The "today" file is one of the most commonly used files as we captured its usage on 11 of the 15 days that we obtained data. The interactions take place in a wide variety of situations and tend to be brief with intermittent usage throughout the day. In the midst

of other tasks, the user will quickly switch to this file to jot down an item or check the list. Likewise, he will occasionally browse the file to review the list more thoroughly and remove old items.

The contents of this file are terse notes to the user that serve as reminders. These are often simple and can be as short as a one word prompt such as "tax" shown on the second line in Figure 3. The user characterized this file in jest as "everything I should be doing but don't."

2.4.2 Recurring Meetings

The wearable is also used to support recurring meetings. These meetings tend to be one–on–one or at most with a few people where the user is familiar with the attendees and their work. The topics of discussion include new points of interest as well as revisiting old items. While listening and participating in a discussion, the user takes concise notes on general points of interest or specific details that he wants to remember. The focus of the user's attention is on the discussion, but the user takes notes as a background task.

Terse notes are sufficient because they tend to be accessed only in the context of the

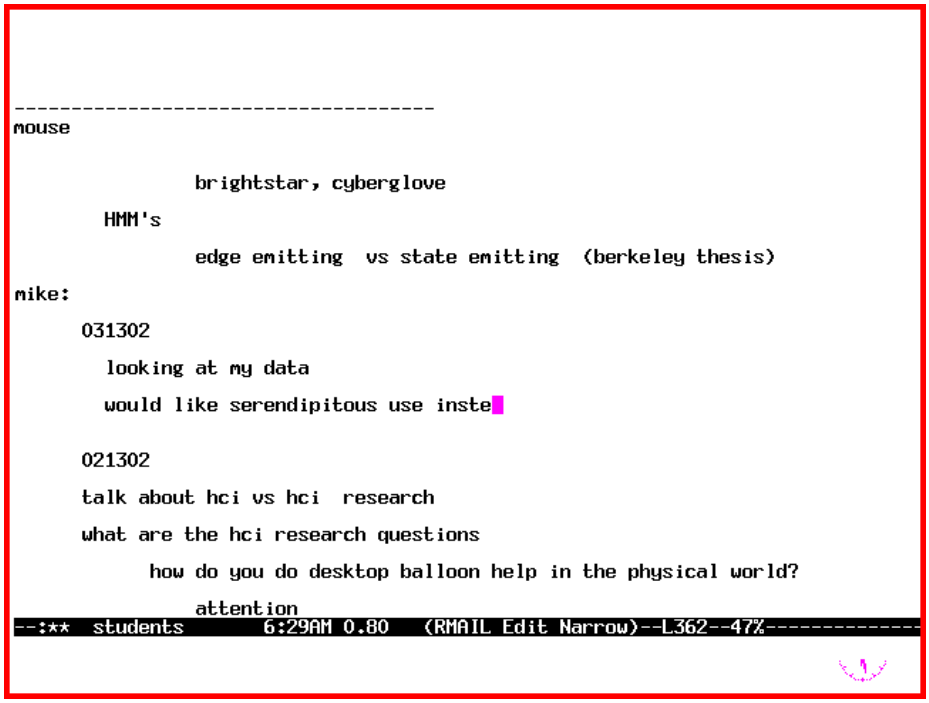

Figure 4: An example of notes from a recurring meeting.

meeting, whereas other styles of notes are accessed outside the context in which they were taken. The user's organization of this type of information affords quick review of notes from past meetings. This interaction is fast because the previous notes are stored adjacent to the new ones, and as the user is often already wearing the display, the only operation needed to peruse the file is paging up or down. We discuss the user's organizational schemes in more detail in Section 2.5.2.

This note–taking practice occurs regularly given that meetings are a common work activity for this user. A typical example is shown in Figure 4. This is a screen shot captured during a one–on–one meeting a few minutes after the start of the conversation. Here is a reconstruction of the interaction with the wearable that took place during this meeting:

> The user first opened the file of all notes on student meetings called "students" (see status line on Emacs buffer). He found the proper place to record new notes about the conversation with this individual by searching for his name, "mike". He created a new spot for this meeting by entering a few blank lines between the name and the previous meetings notes which start with the line containing "021302". Next, he typed in a the string "031302" representing March 13th 2002, and a few lines of text that are notes on the current conversation. The user continued to take a few lines of notes for the duration of the meeting.

The area of the file for this person is marked with the line "mike:". The user said that he uses this convention of a name followed by a colon to attribute some information to a person. Here, the mark is used to attribute the notes to Mike and can later be used as a key when searching for this information. Following that line are subsections for meetings from different days with that person. Each of these begins with strings representing the date and is followed by the notes from that meeting. The text before "mike:" is notes taken during a conversation with a different person.

2.4.3 Talks and Demonstrations

Over the course of our study, our participant attended several events relevant to his interests such as talks and demonstrations. These activities tended to be one–time meetings with a single person disseminating information. The speaker was often from another institution and usually had infrequent contact with the user outside this event. Talks are often given in a class room or meeting room with the wearable user sitting in the audience. When the user is attending a demonstration, there are usually only a few other people listening to the speaker at one time, and often there are many other demonstrations going on. The user will often walk up to one demonstration, listen and take notes for a few minutes, and then go to another demonstration.

In this setting, the user generates more descriptive and complete notes compared to the previous examples. He stated his goal as "want[ing] to refer back to research notes" whenever they might be relevant. For a talk that the user attends, he creates a new file in the "talks" directory and names the file after the speaker's last name (for example "Tern" in Figure 5). For a demonstration at a remote site, a new section of an existing file about

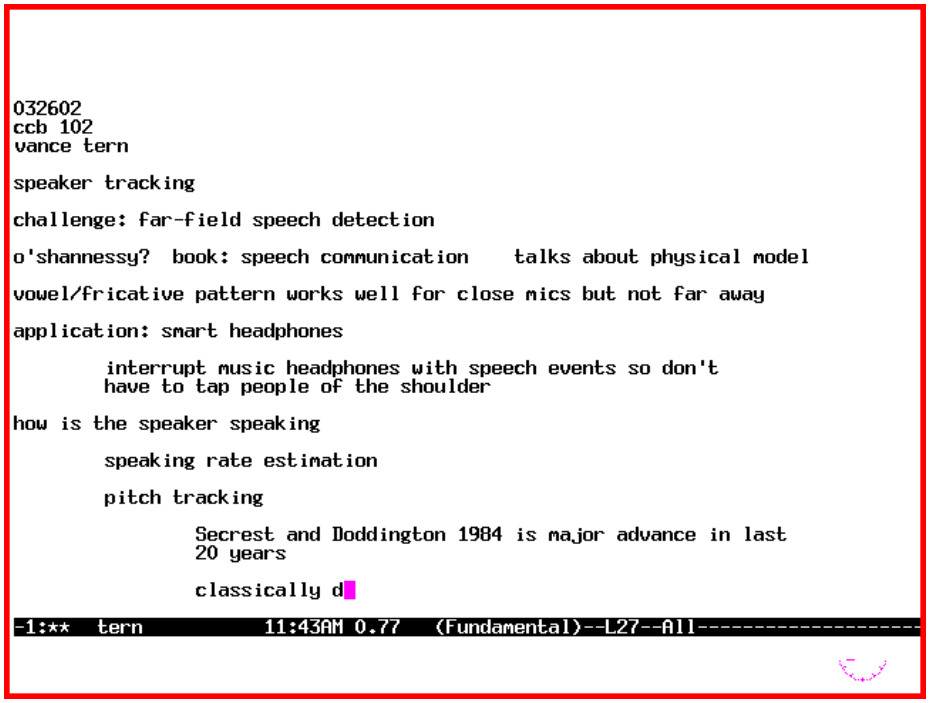

Figure 5: Research notes taken during a talk.

the place or trip is made or, if needed, a new file is created.

The notes begin with some basic context about the situation, usually including the date, person, and location. Tabs and new lines are used to separate and organize the ideas represented in the notes (Figure 5). The user actively structures, restructures, and fills in more details as the talk progresses and his understanding of the content changes.

2.4.4 Contact Management

The wearable user has a file named "phones" devoted to contacts. He uses this file to help manage information about the people he knows. It contains information such as names, phone numbers, email addresses, titles, and locations or addresses of people. In addition to this traditional contact information, the user includes other reminders about the person which are stored in the same file. He often has a note about when he last met the person and why they met, or a more general description of why that person is included in the file. Directions to locations are not uncommon and sometimes include travel times.

The user indicated that he would write down new information when he met someone. Usually, this new information came from a business card. Our data showed that the user also occasionally copied email signatures from messages stored on the machine into the file.

This file is used when the user wants to remember details about a person he just met or to recall information about someone he met previously. When he encounters someone he has met before, the user quickly searches through the file during the conversation to find when they last met and other information about that person's work.

In Figure 6, we see the variety of ways the user records information about his contacts. For example, the first line was entered because the user frequents a local sandwich shop and repeatedly sees the same employee. However, he can never remember his name, Yan. One time, the wearable user asked for the employee's name and wrote it down at the top of this file so he could look it up the next time he was there. The information for Mara Wareall, Mark Tersey and Dakis Yahonce was all entered while the user was organizing a business dinner. He went through the "phones" file ensuring that he knew how to contact all of the people attending. He did not have any information for these people in the file so he added

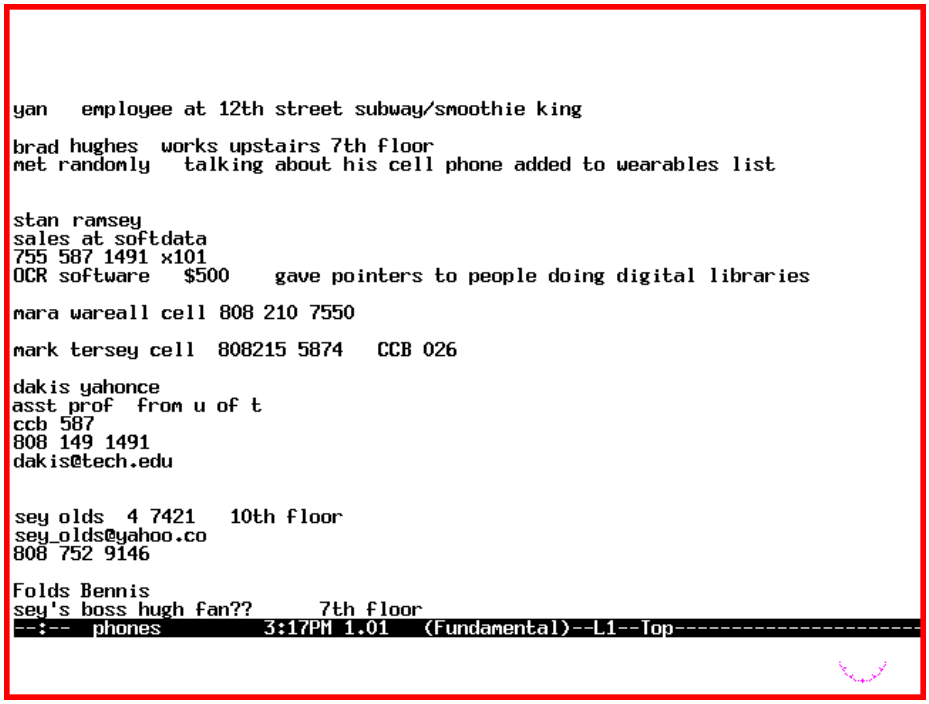

Figure 6: The "phones" file which serves as the user's contact list.

it.

2.4.5 Scratch Pad

While the previous examples occurred regularly, there are other interactions that are less typical but demonstrate the versatility of the user and his wearable computer. One instance is the computer's use as a scratch pad (Figure: 7):

> After a pause in using the machine of about ten minutes, the user was at a command line prompt in an xterm. He cleared the screen and started entering a string of numbers at a moderate rate: "1 3 2 4.5 3 6.75 4 1...". The input was obviously not a command to be executed. After 43 seconds, a total of 10 numbers were entered. Then there was a pause of 13 minutes after which the user continued use of the machine by first closing the xterm, erasing the numbers.

When queried about the purpose of the numbers in an interview session, the user indicated that he was doing some math in his head and was writing down the intermediate

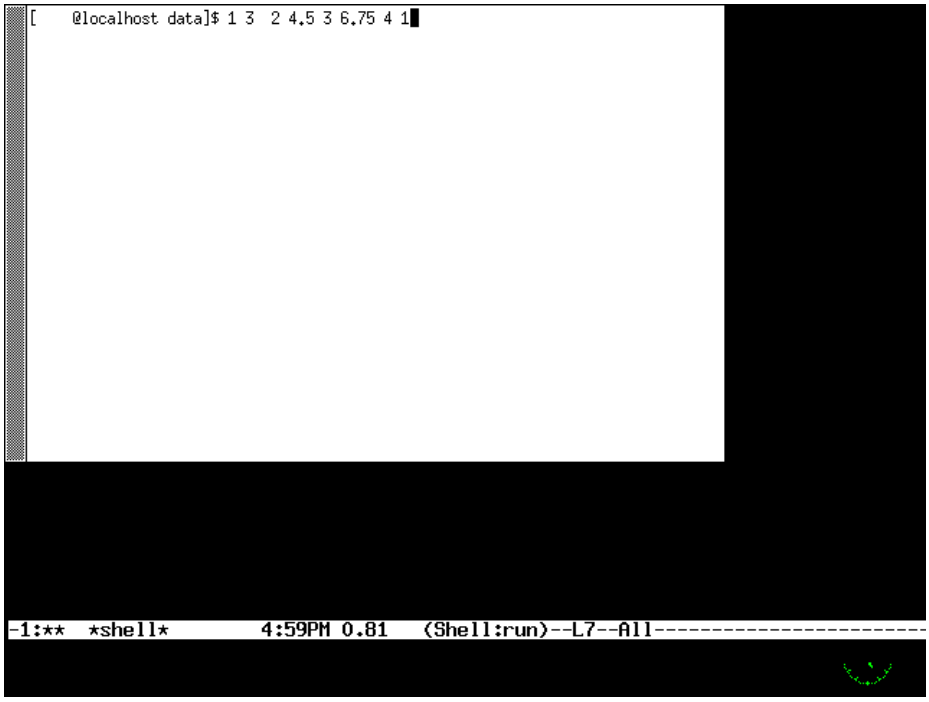

Figure 7: Using the wearable as a scratch pad to jot down numbers.

results. He happened to use the xterm that was available on screen. He did not want to worry about opening a file or saving the information and just needed to jot down some numbers.

The user's own working memory or a scratch piece of paper could have sufficed, but the wearable provided adequate support for this type of task. He was able to use the wearable as a scratch pad since there was very little setup time. The machine was most likely more convenient than looking for a piece of paper since his machine is always with him and has been integrated into his way of working.

2.5 Discussion

These examples of "today", recurring meetings, talks and demonstrations, contact management, and scratch pad show several main trends: the wearable as a device to augment the user's memory, the importance of the user's information organization scheme and how the wearable is often used as an aid to a social task. These usage patterns highlight the versatility of the wearable computer and the strategies adopted by the user to enable effective use in everyday situations.

2.5.1 Memory Augmentation

A key theme of the wearable's use supported by our data collection and exemplified in the previous section is how the user has adopted the wearable computer as a tool to augment his memory. The machine is employed to aid the user's memory over a spectrum of time frames and in a large variety of situations. There is a low cost associated with machine use since the machine is almost always with the user, the interaction is quick using a head–up display, and the user has a rapid means for text entry with the Twiddler. Together, the user leverages these features to store information in his self–described "other brain."

The majority of interactions with the machine augment the user's long–term memory in some way. The user relies on the machine's perfect storage capability to compensate for the fact that his memories can degrade with time. The "today" file is used to remember near–term events. The meeting notes serve as reminders in the context of the meeting about past discussions. The "phones" file archives a variety of information about whom the user has met. Lastly, notes from talks and demonstrations comprise a large amount of archived information relevant to the user's work.

On a few occasions, the data also revealed the user applying the wearable as a tool for short–term memory augmentation. These interactions are characterized by the need to remember a small number of items for no more than a few minutes. The previous scratch pad example demonstrates this technique. The user employed the wearable because it was a convenient place to jot down some numbers while performing calculations in his head. Instead of remembering the temporary values or finding some other support mechanism, the wearable computer interaction was fast enough and the machine flexible enough to aid the user. Similar to how working memory is used, the items are temporary, and there is no need for long term storage.

While the user employs the wearable computer for augmented memory support, it does not replace the user's memory. Instead, it serves as a repository for details in which the notes provide cues to refresh the user's memory.

2.5.2 Information Organization

The data show that the user has developed an intricate scheme for organizing his information space. The user is able to quickly write down information and then navigate his information space with his keyboard and head–up display. In addition to using traditional file hierarchies, there is often structure within individual files. The notes from a meeting (Figure 4) represent a composite file consisting of several separate entries from meetings with different people on different days. Another example of this technique includes taking notes in a file containing several emails on a subject. In general, tabs, blank lines, dashed lines, or email headers are used to define the structure within a file.

Within a composite file, the user can impose additional structure to keep related information together. In the "phones" file, the user indicated that he often tries to group people from the same organization together (Figure 6). The student notes file is organized by person at the highest level. Each area devoted to an individual is further subdivided into meetings labeled with the date (Figure 4).

There were only a small number of explicit retrievals found in the data; however, the composite file structure might facilitate incidental access. Because the user co-locates related information he can quickly and easily review previous notes as he is about to enter new ones.

2.5.3 Wearable in support of Conversations

While the machine is commonly used to augment the user's memory, most of the interaction occurs under tight attention or time constraints because the user is actively involved in some other primary task. In particular, our data reveal that most of the situations the wearable is used in are social. In a conversation, the user might take notes on points of interest or retrieve support material from the machine relevant to the discussion. However, the primary focus is still on the conversation at hand, and the user tries to adhere to the social constraints of the situation.

While engaged in another activity, the user must quickly make many decisions that govern his interaction with the machine. First, to use the machine effectively for memory

augmentation, the user must know where to store new notes or find old information. As mentioned above, the user has developed several strategies that revolve around the organization of his information which enable him to quickly return to the task at hand.

While taking notes, the user also decides how much effort and time to spend on recording the information. For a subject familiar to the user, he may only record details that he might otherwise forget such as an action item from a weekly meeting (Figure 4). For less familiar material of interest, he might spend more time taking richer notes (Figure 5). The process of recording the information with the wearable computer tends to take minimal attention as the user touch types his notes at a rapid pace on the Twiddler keyboard, and the head–up display enables him to check on the notes being written with a quick glance.

Even while primarily engaged in a social activity, it is clear from the data that the user does occasionally shift his focus to the machine while recording information. This usually takes the form of editing the content of the notes or restructuring them. On several occasions, the user would go back a few lines and change a line of text or expand on an idea by typing more details. The user indicated he changed the structure of the notes so he could facilitate the access of information when needed. Furthermore, he said that if he did not spend the time to organize the information while note taking, he knew he would not go back later to do so.

On some occasions, the user spends time directly interacting with the machine. These interactions usually center around maintenance of his information. There might be other people around, but he is not engaged in activity with them. For example, although the user generally decides where to place information as he is storing it, he sometimes explicitly spends time consolidating and organizing his data. While the user was on a trip and preparing to meet his hosts and attend a demonstration, he spent part of the morning going through a collection of email he had gathered about that trip. He reviewed his email and copied contact information into the "phones" file. He annotated and rearranged the information so he could refer to it later that day when he met his hosts. He spent time before the meeting, so that when he was engaged in the social situation the information he wanted was readily at hand.

It is worth reiterating that none of the applications our participant used during our study were designed specifically for wearable computers. The current machine and programs are sufficiently flexible to enable this expert user to operate in these conditions with the aid of his strategies.

2.6 Conclusions

This case study provides a preliminary understanding of some of the capabilities a wearable computer can provide in supporting everyday life. We found that the wearable computer was used by the expert to aid his memory in a large variety of situations. Furthermore, the user has developed several strategies that enable him to use the wearable computer in situations where his attention is limited. The wearable is occasionally the primary focus of attention; however, it is also common for the machine to be used in a secondary role supporting the conversations of the user.

It is difficult to generalize these results given the small sample size of one user; however, the data does provide interesting insight for mobile computing. Our study shows use of the wearable in many situations where other researchers have shown problems in the adoption of mobile technology [32, 9, 21, 5]. In particular, the wearable is used to store a great number of small pieces of information (in addition to many large pieces) and is used in support of many daily activities such as conversations.

One key difference between traditional mobile technologies and the wearable computer from this study is the user's ability to enter information during conversational situations with the Twiddler keyboard. In the rest of this dissertation, we explore input mechanisms that can be used in conversational situations like those demonstrated in our case study.

CHAPTER 3

THE TWIDDLER ONE-HANDED CHORDING KEYBOARD

The expert from our case study and several other wearable computer users [35, 57] have adopted the HandyKey Twiddler (Figure 8), a mobile one–handed chording keyboard. From our experience with the Twiddler, we have anecdotal evidence that an expert user can type rapidly and use the keyboard to enter information while engaged in everyday tasks.

More generally, mobile typing is becoming increasingly important. With 1.3 billion users, mobile phones have become ubiquitous in many parts of the world [3]. Similarly, the use of wireless text messaging is becoming widespread with predictions of a rate of over 1 trillion messages per year being reached shortly [33, 42]. These statistics are remarkable considering the inefficiencies and poor design of current text entry methods for mobile devices.

Increasing text entry rates has a long history, and recently there has been a resurgence in research on physical keyboards exploring how they can be used for mobile devices. Improving text entry speed may open new markets for wireless email, which is desired by 81% of consumers in one survey [12], and wireless email is predicted to drive the next stage of the industry's European data revenues [13]. Unexpected segments of the user population may benefit from improved text entry capabilities. For example, many in the Deaf community have adopted wireless texting as a convenient means of communication.

In the following three chapters, we present our research on the Twiddler keyboard. In the current chapter we describe the Twiddler keyboard and how it compares to typing on similar 3x4 keypads of mobile phones. We then present a longitudinal study comparing the learning rates for the Twiddler relative to the *de facto* standard for mobile phone text entry, multi–tap. In Chapter 4 we present a continuation of our study designed to explore expert characteristics of Twiddler typing, and in Chapter 5 we explore how to improve a novice

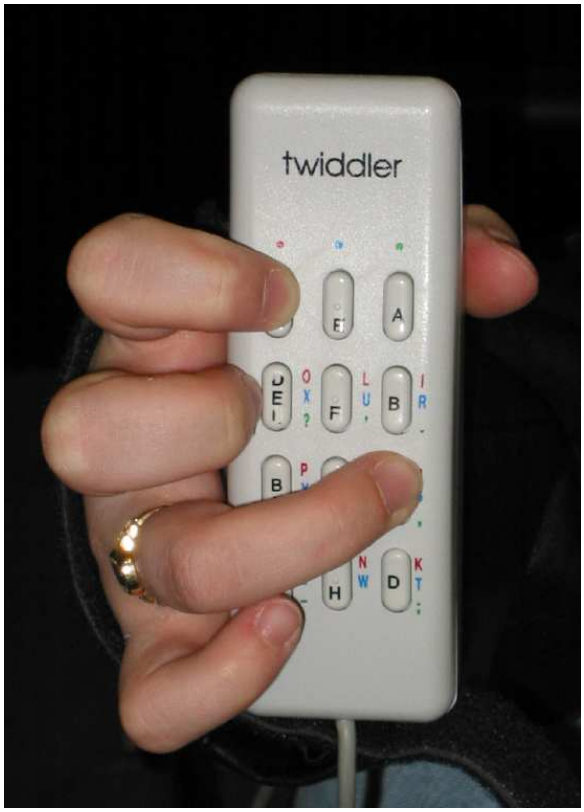

Figure 8: Chord for the letter 'j' (R0L0) on the Twiddler

Twiddler user's typing experience.

3.1 The Twiddler Keyboard

The Twiddler is a mobile one–handed chording keyboard with a keypad similar to a mobile phone (Figure 9). It has twelve keys arranged in a grid with three columns and four rows on the front. Unlike a mobile phone, the Twiddler is held with the keypad facing away from the user, and each row of keys is operated by one of the user's four fingers. Instead of pressing keys in sequence to produce a character, multiple keys can be pressed simultaneously to generate a chord. Additionally, the Twiddler has several modifier buttons such as 'Alt', 'Shift', 'Control', etc. on the top–back operated by the user's thumb (Figures 10 and 11).

The default keymap for the Twiddler consists of single button and two button chords which are assigned in an alphabetical order and is divided into three parts. Characters 'a'–'h' only require one button press ("single"). The letters 'i'–'z' are typed with chords of two buttons. For these letters, two of the buttons on the top row act as shift keys. The

Figure 9: The Twiddler next to the Sony Ericsson T610 mobile phone.

Figure 10: The Twiddler from different angles.

shift button for 'i'-'q' is called the red shift, and the shift for 'r'–'z' is the blue shift. This nomenclature is derived from the keymap printed on the face of the Twiddler.

The default keymap for the Twiddler is shown in Figure 12. The four characters in the Buttons column denote what keys to press from each row. 'L' indicates the leftmost button in a row, 'M' the middle and 'R' the right button. A '0' means the corresponding finger is not used in the chord. Note that the designation for left and right is from the user's perspective while holding the keypad facing away. As a result, there is a left–to–right mirror between Figure 12 and Figure 8. Figure 15 shows the representation of the chording layout from the user's perspective.

For example, the chord for 'a' is 'L000' which indicates that a user presses the left button on the top row from her perspective. To generate 'j' ('R0L0'), a user presses the right key on the top row and the left key on the third row (Figure 8).

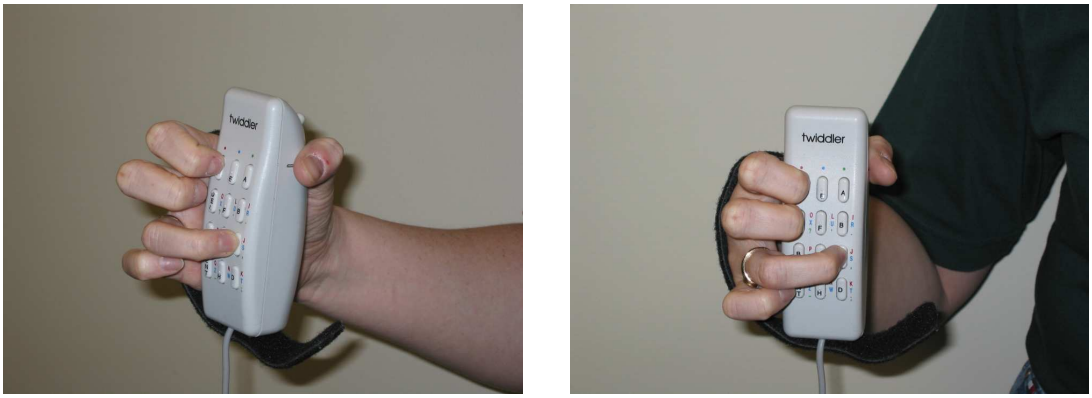

Figure 11: The Twiddler being held in typing position.

With traditional keyboards, a character is generated when the corresponding button is pressed. This strategy cannot be used for chording since the user may not press all of the keys for the chord at exactly the same time. Instead, the Twiddler generates the keycode once the first button of a chord is released. Just before this point, all of the buttons for the chord have been depressed so the proper keycode can be generated. In Section 4.2, we explore the relationship between the timings of pressing the buttons and how they relate to learning to chord.

For a chord on the Twiddler, each of the fingers may be in one of four states (pressing one of three buttons, or not pressing anything). Ignoring the "chord" in which no buttons are pressed, there are $4^4 - 1 = 255$ possible chords using the four main fingers. The modifier buttons operated by the thumb allow more chords. HandyKey includes what we have termed multi–character chords (MCCs) in the default keymap: single chords that generate a sequence of several characters. For instance, there are chords for some frequent words and letter combinations such as 'and ', 'the', and 'ing '. Users can also define their own MCCs. We present an evaluation and analysis of the effects of MCCs on expert typing rates in Section 4.3.1.

3.2 Typing on Mobile Phone Keypads

There are two ways to accommodate the small form-factor keyboards that are resulting from the decrease in size of mobile technology: make the keys very small, like on mini-QWERTY keyboards, or remove the one–to–one mapping between keys and characters. Most phones

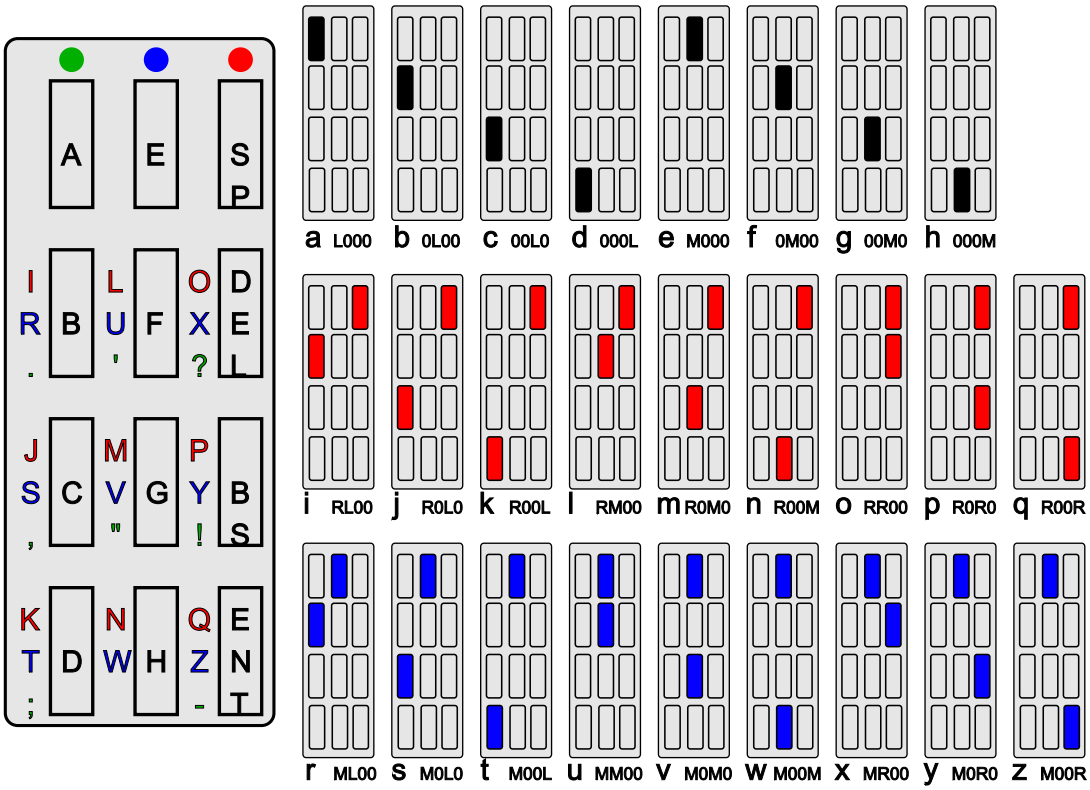

Figure 12: Keymap for chording on the Twiddler.

map more than one character onto a key because they inherited the 12 button keypad of traditional phones. When multiple characters are assigned to one key, a method is needed to disambiguate between the possible options. Wigdor and Balakrishnan [66] present a taxonomy with three dimensions for ways to disambiguate: the number of keys used (one or more), the number of presses performed on the key(s) and the possible temporal ordering of key presses (consecutive or concurrent). These methods can be further combined with linguistic models to disambiguate the key presses. Chording on the Twiddler represents a point in this space which uses concurrent presses from multiple buttons.

For mobile phones, multi–tap is a very common text entry technique. The alphabet is mapped onto 8 of the 12 buttons on the mobile phone keypad resulting in 3 to 4 letters per key. To generate a character and disambiguate between the characters on the same key, the user presses a single key multiple times to cycle through the letters until the desired one appears on the screen. Users hold the keypad towards them and can enter text with

one or two hands using one or two fingers/thumbs. Once the desired letter appears, users can press the next key to start the process again for the next letter, use a kill key function, or wait for the timeout. The timeout is a feature that deactivates the current key after a specified amount of time.

Because multi–tap is so prevalent on mobile phones, it has become the *de facto* baseline for which to compare other mobile phone entry techniques. Research has found multi–tap typing rates for novice users ranging from 7.2–8.7 wpm with 15–30 minutes of practice [38, 65, 66, 36]. These studies show that as users gain experience their typing rates can increase to 11.0–19.8 wpm. Other research has predicted maximum expert typing rates of 20 to 27 wpm [54]. Although multi–tap is very common typing method, it is also relatively slow.

T9 is another common mobile phone input method. Like multi–tap, multiple letters are assigned to each button on the keypad. However instead of the user disambiguating every character with multiple button presses, T9 uses language disambiguation. Using a dictionary, T9 presents the most probable string the user is attempting to enter given the input so far. If the presented text is incorrect, the user can press a special key to cycle through possible alternatives. One study found that novice users type 9.1 wpm while experts can achieve 20.4 wpm [26]. Unfortunately, T9 rates drop drastically once the user needs to enter words that are not in the dictionary, such as proper nouns.

Recently several new methods have been developed for entering text on mobile phone keypads including LetterWise [38], TiltText [65], and ChordTap [66]. These methods offer novice performance similar to multi–tap (7.3 wpm, 7.4 wpm and 8.5 wpm respectively). In addition, each of these methods offers faster expert typing rates than multi–tap given the same amount of practice. LetterWise users achieved a rate of 21 wpm after approximately 550 minutes of practice. TiltText users reached 13.6 wpm and ChordTap 16.1 wpm respectively with about 160 minutes of typing practice.

Table 1 provides a summary of this work and also includes the results of the studies performed in this chapter and Chapter 4. Where it could be derived, the experience column shows the approximate number of minutes the novice user spent typing with the given

method before the maximum words per minute were calculated. Studies that were not longitudinal in nature but characterized subjects as "novice" or "expert" are marked accordingly. In summary, these studies reveal that most text entry methods are much slower than the rates we will show are attainable with the Twiddler.

Table 1: Comparison of mobile text entry rates using 3x4 keypads.

Method	**Keyboard**	**Experience**	**WPM**
Chording	Twiddler	1580 min	67.1[1]
Chording	Twiddler	1500 min	47.1
Chording	Twiddler	400 min	26.2
LetterWise [38]	desktop keypad	550 min	21.0
T9 [26]	Nokia 3210 phone	expert	20.36
Multi-tap	Twiddler	400 min	19.8
ChordTap [66]	modified Motorola i95cl phone	160 min	16.1
Multi–tap [38]	desktop keypad	550 min	15.5
TiltText [65]	modified Motorola i95cl phone	160 min	13.6
Multi–tap [65]	Motorola i95cl phone	160 min	11.04
T9 [26]	Nokia 3210 phone	novice	9.09
Multi–tap [26]	Nokia 3210 phone	novice	7.98
Multi–tap [26]	Nokia 3210 phone	expert	7.93
Multi–tap [8]	desktop keypad	n/a	7.2
Two key [8]	desktop keypad	n/a	5.5

3.3 Experiment: Chording versus Multi–tap

We present our longitudinal experiment comparing chording to multi–tap. Ten subjects participated in 20 sessions over the course of three weeks where each session lasted approximately 45 minutes. Each session consisted of typing text phrases in both conditions and included a 5 minute break. Depending on the condition under test, the testing software presented the participants with the keyboard layout for either multi–tap or chording and

[1]Typing rate of fastest participant from study.

statistics of performance. A phrase was presented on the screen and users transcribed the text with the current input method (Figure 16).

3.3.1 Participants

Twelve participants were recruited from the Institute's student body. All of the subjects were informed of the significant time commitment required for the study and were compensated for their participation calculated at the rate of $1 x WPM x Accuracy over the entire session, with a minimum of $8 per session. Two participants dropped out within 8 sessions due to time constraints. Of the ten subjects that completed the study, eight are male and nine right–handed.

Eight of the participants reported that they owned or used a mobile phone on a regular basis, and none of the subjects had used a Twiddler before this study. We chose only native English speakers as our test phrases were in English. We also recruited participants without long fingernails that might have impeded typing speed.

3.3.2 Equipment and Software

The experiment was conducted in the College's usability laboratory. This was a stationary test where participants sat at a computer running our test software developed in Java. The computer stations were Pentium III based PCs. The Twiddler was attached to the computer via a serial cable and continually sent the state of all of its buttons to the computer at 2400 baud, resulting in a key sample rate of approximately 45Hz. The software parsed the serial stream as text input.

The faceplates of the three Twiddlers used for this study were modified with labels for multi–tap (Figure 13). Labels are appropriate since multi–tap is designed to be used while the keypad is facing the user; however, when chording, the Twiddler keypad faces away from the user. To prevent subjects from turning the chording keypad to look at the keys, the chording labels on the Twiddler were covered. The labels also posed another potential problem due to left and right mappings (as discussed in Section 3.1). The test software displays key presses to the user as if the Twiddler were held as intended. If the participants turned the keypad around for the chording condition, they would have to mirror the image

Figure 13: On the left, typing using multi–tap on the Twiddler keypad. On the right, chording with the Twiddler one–handed keyboard.

in their heads.

3.3.3 Design

The experiment is a 2 x 20 within–subjects factorial design and is similar to previous text entry research designed to determine the learning rates of different typing methods using longitudinal studies [40, 38].

We presented the participants with two conditions: multi–tap and chording during 20 sessions over the course of three weeks. Sessions were scheduled Monday through Friday where each session was separated by at least two hours and no more than two days. Each session lasted approximately 45 minutes and consisted of two parts delineated by typing condition. Participants were randomly assigned to a condition (balanced across participants) for the first session. This condition was tested first followed by the second condition. The order of presentation alternated from session to session.

Depending on the condition under test, the testing software presented the participants with the key layout for either multi–tap (Figure 14) or chording (Figure 15) and statistics

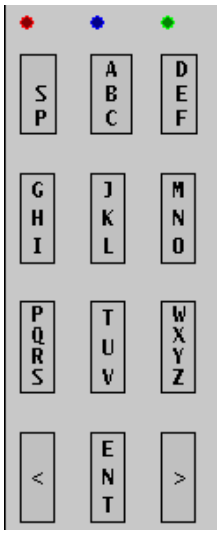

Figure 14: Layout for multi–tap.

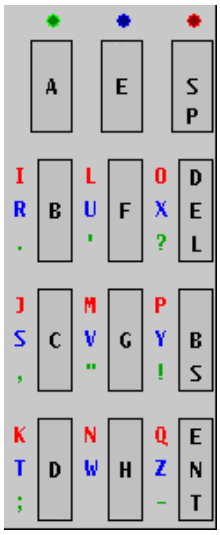

Figure 15: Layout for chording.

of performance. A phrase was presented on the screen above the transcription that resulted from the subject's key presses (Figure 16).

Each half session began with a warm–up round that was not used in measuring performance. The warm–up consists of typing the two phrases, "abcd efgh ijkl" "mnop qrst uvwx yz" twice. During the warm–up phase the program also highlights the correct buttons to press to type the next letter in the phrase. Once the warm–up phase ended, the highlighting was turned off, but the key layout remained. The subjects were instructed to begin typing for the trials, and data recording began.

Each half session consisted of several blocks of trials. Each block contained ten text phrases of approximately 28 characters each and were selected randomly from the set of 500 phrases developed by MacKenzie and Soukoreff [39]. These phrases are specifically designed as representative samples of the English language. The phrases contain only letters and spaces, and we altered the phrases to use only lower case and American English spellings.

The experimental software presented blocks of phrases until twenty minutes had expired. As participants' typing rates increased throughout the study, the number of blocks completed also increased. In the first session, participants typed 5 to 8 blocks total and completed 12 to 21 blocks by the final session.

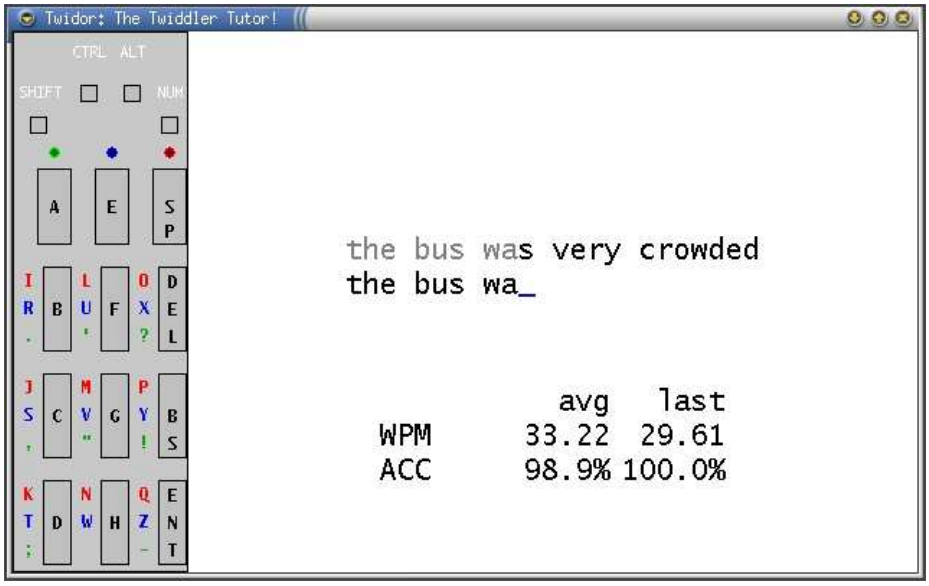

Figure 16: Experimental software showing the chording layout, phrase and statistics.

During the first and last sessions, we also asked each participant to type using a standard desktop QWERTY keyboard for two blocks for a total of 40 phrases. We collected this data as a baseline typing rate for each participant.

The software collected data at the level of button presses. Every key press and release is recorded to a log file. When a button is pressed or released, the system logs the time–stamp (obtained with Java's *System.currentTimeMillis()* system call), the character generated (if any), and the state of all of the Twiddler's buttons. The current text entry method is logged as well as the phrases presented to the participant. With this data we can determine when each key was pressed and released, the duration each button was held, the time between releasing one button and pressing the next, and the resulting transcribed text.

3.3.4 Procedure

Each participant was given written, verbal, and visual instructions explaining the task and goal of the experiment. The researcher explained how to type for both methods on the Twiddler and demonstrated how to hold the device for each condition. He also explained that the key layout mimics a mobile phone, mapping number keys to Twiddler keys. Finally, he showed the participants how to press each letter of the alphabet for both methods. For multi–tap, he explained that the keypad is held facing the participants. The participants

were informed they could wait for the timeout or utilize the kill button, and they could use one or two index fingers/thumbs to type. For chording, the researcher showed the participants how to strap the Twiddler onto their hand. He also showed how to press each key with the tip of the finger and how to press multiple keys simultaneously to generate chords.

The software was self–administered (under researcher supervision), and participants had unique anonymous log–in IDs. The subjects were asked to copy a presented phrase by typing on the Twiddler keyboard. They were instructed to type as quickly as possible while minimizing errors. The program provided statistical data as feedback so the participants could monitor their progress. In addition to the phrase to be typed and the statistics, the program also displayed the keyboard layout for the current method for reference.

Once started, the program initiated the warm–up phase for the appropriate method. Once the four warm–up phrases were typed, the program instructed the participants that the timed trials would start. After each block of ten phrases, the program paused to show the participant's statistics of rate and accuracy for that block. After 20 minutes, the program displayed the statistics for that half of the session and instructed the participant to take a five minute break. After the break, the program switched to the second input method. The participant changed grip on the Twiddler to be compatible with the method, and the second half of the session proceeded like the first.

3.4 Results

For each of our ten participants, we collected approximately 2100 transcribed phrases. In total for both conditions over all 20 sessions and 10 users we collected 600,000 transcribed characters.

3.4.1 Text Entry Speeds and Learning Curves

The mean entry rates for session one were 8.2 wpm for multi–tap and 4.3 wpm for chording. As sessions continued, the means improved and reached 19.8 wpm for multi–tap and 26.2 wpm for chording by session 20. While both showed improvement, the performance scores for the chording condition rapidly surpassed those of multi–tap (Figure 17).

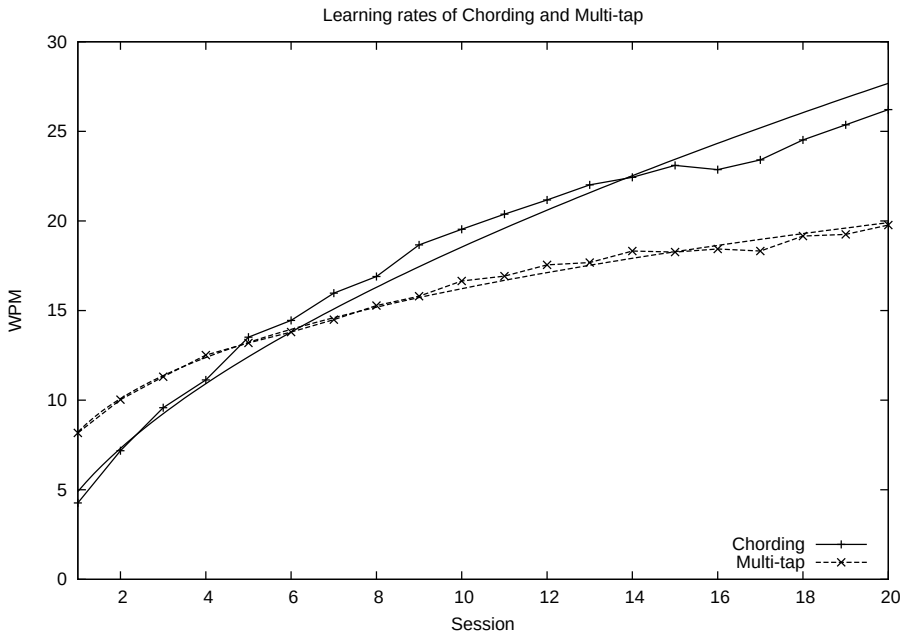

Figure 17: Learning rates and exponential regression curves for multi–tap and chording for 20 sessions.

An analysis of variance (ANOVA) of text entry speed shows a main effect for typing method ($F_{1,9} = 45.2, p < 0.0001$) and for session ($F_{19,171} = 36.8, p < 0.0001$). There is also a significant method–by–session interaction ($F_{19,171} = 3.62, p < 0.0001$).

The main effect of session was expected as was the method–by–session interaction. The participants learned to type faster over the course of the 20 sessions. Initially the participants on average typed faster with multi–tap, but after a few sessions the difference eroded and by the eighth session chording became faster ($T_9 = 3.1, p < 0.05$). The magnitude of the differences also increased as the sessions continued.

For each typing method, we derived exponential regression curves to model the power law of practice (Figure 17) [10]. The equations for the curves are below. The x values are the number of 20 minute sessions and the y values are the predicted rate in words per minute for that session. The curves have R^2 values greater than 98% indicating that the curves are fit the data well, accounting for over 98% of the variance. As can be seen, multi–tap rates begin to plateau while the chording method shows steadily increasing typing speeds.

$$\text{Twiddler: } y = 4.8987x^{0.5781}, R^2 = 0.9849$$

$$\text{Multi–tap: } y = 8.2235x^{0.2950}, R^2 = 0.9961$$

The crossover point in the curves indicates where one condition's typing rate surpasses the other. In our study, the chording method began with slower speeds but quickly overcame multi–tap. The crossover occurred after the fifth session or after 100 minutes of practice.

3.4.2 Per Participant Text Entry Rates

Because learning rates are exponential, we graphed the text entry rates per participant as a log-log plot. Both graphs in Figure 18 show data for all ten subjects on a per session basis. The left side of Figure 18 shows the chording data and the right is for multi–tap. The steep slope of chording indicates rapid learning. The slopes of the multi–tap sessions are much more shallow. The curves also show the large variances in the multi–tap entry rates.

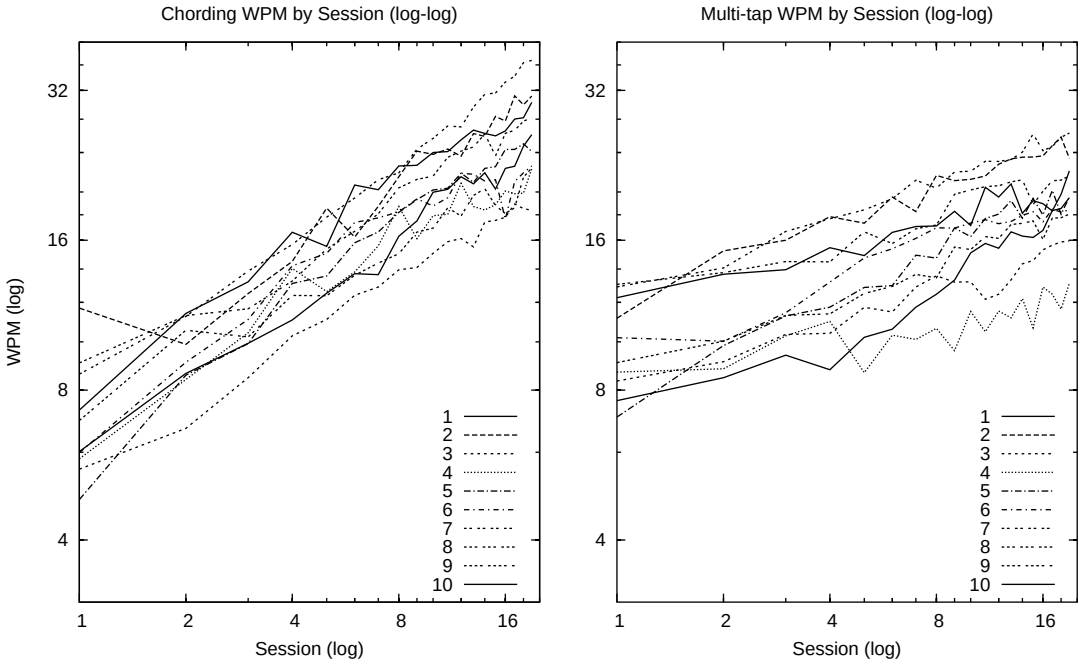

Figure 18: Log-log plots of learning rates for chording (left) and multi–tap (right) for each participant.

Figure 19 illustrates per–user regression lines that model per–phrase typing rates for the chording condition. In our 20 sessions, each participant typed approximately 1050 phrases for each condition. We have extended the regression lines to predict the rates experts might achieve. The chording regressions are particularly interesting because of the clusters that

appear. It suggests that the faster typists would reach 60 wpm, the rate of our expert, after 10,000 phrases (approximately 80 sessions or 27 hours) while the slower typists could achieve 45 wpm. We will explore these predictions more thoroughly in Chapter 4.

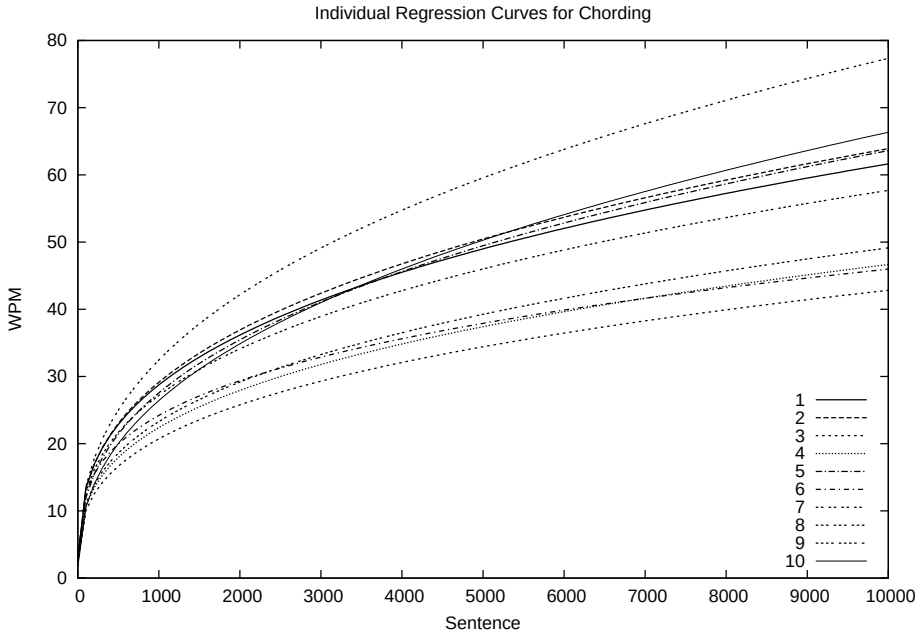

Figure 19: Per participant regressions for chording.

3.4.3 Error Rates

We used Soukoreff's and Mackenzie's total error rate metric [56] which combines corrected and uncorrected errors. Our participants tended not to correct their mistakes, so most of the errors in this study remained uncorrected.

Figure 20 shows the average total error rates per session for both conditions. Our error rates are comparable to other studies [38], and all of the error rates are less than 5% after the second session. The chording method error rates started at 10.4% but quickly decreased. We believe the high initial rate is due to the fact that the participants had no experience with chording on the Twiddler. The gradual upward trend in error rates is likely an artifact of our compensation scheme. Our participants learned they could earn more money by trading off some accuracy for a slightly faster typing rate.

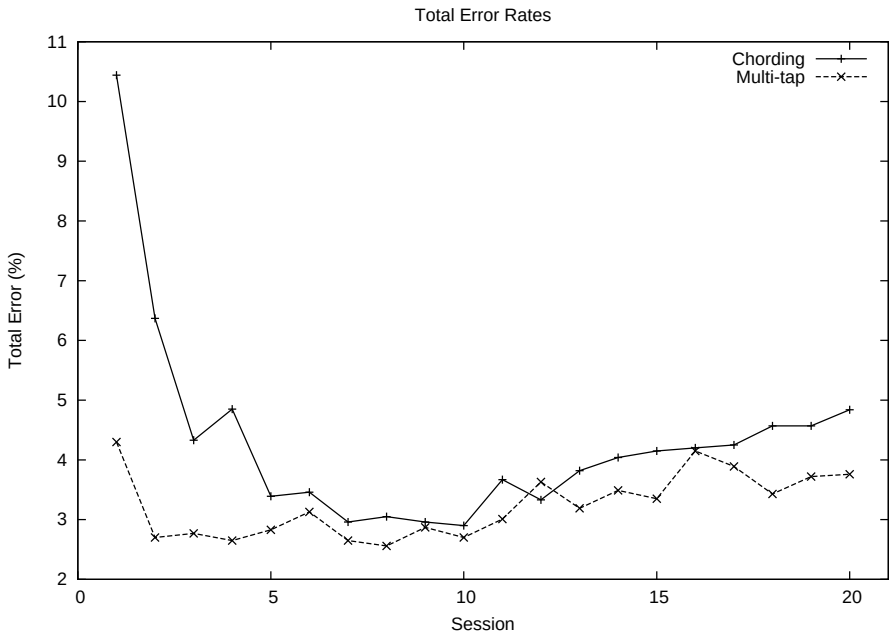

Figure 20: Total error rates for chording and multi–tap.

3.5 Discussion

3.5.1 Multi–tap Typing Rates

Our study data for multi–tap reveal a wide range of values for typing rates across users (Figure 18). One explanation for this is prior knowledge and experience with multi–tap. All but two subjects reported owning mobile phones. Multi–tap is a common technology, and it is hypothesized that participants are familiar with how it works even if they do not use it often. At the very least, the mapping of letters to numbers on a phone is familiar. This might also be a reason for multi–tap rates beginning higher than chording. Another factor in multi–tap's initial advantage is participants' lack of experience with chording. All reported never using chording or the Twiddler before.

Our study also reveals higher typing rates for our participants than previous studies. The James and Reischel study [26] found multi–tap typing rates of 8 wpm; our participants started close to this rate (8.2 wpm) but quickly surpassed it. One possible explanation for this increase is that while James and Reischel's subjects may have been experienced with sending text messages, they may not have had as much practice as ours. Another possible explanation is the keypad itself. The researchers used a phone keypad while we

used a Twiddler which has larger buttons spaced farther apart. MacKenzie et al. also did not use a mobile phone keypad. Their starting rates were comparable to ours, but our participants' final rates were higher. Perhaps an explanation for this is that we allowed 2 finger/thumb entry. Another factor could be that our participants had a rapid base typing rate on standard QWERTY keyboards (Table 2).

It was also observed that all of our participants touch typed for both methods, looking only at the screen not the keypad. Silfverberg examined the ability to type on keypads with different haptics and found significant effects with varying visual attention [53]. It is possible the Twiddler has better tactile feedback than the phones used in other multi–tap studies.

3.5.2 Comparison of Chording and Multi–tap

As we have shown, novices initially have faster typing rates using multi–tap compared to chording. However, after practice, chording becomes the faster typing method and greatly exceeds the multi–tap rates. Furthermore, our regression lines suggest that the chording method has a greater potential typing rate. With only a little more practice, our participants might achieve typing rates comparable to our expert.

Keystrokes per character (KSPC) is a metric of how many keys need to be pressed for a particular typing method to generate a character [37]. The KSPC for multi–tap is 2.0432 [38]. For chording, only one or two simultaneous key presses are needed to generate a character. Given Soukoreff's digrams [55], this equates to a KSPC value of 1.4764. Fewer key presses are required in chording to generate the same text as compared to multi–tap, thus allowing for faster rates using the chording typing method.

Chording on the Twiddler offers even faster potential typing rates due to multi–character chords. One chord (1 or more simultaneous key presses) can generate multiple characters. For example, the word 'and' can be typed letter-by letter with 4 key presses (1 for 'a', 2 for 'n', 1 for 'd') or 1 chord of 2 simultaneous key presses with the default multi-character chord ('a' and 'h' keys). Key strokes per character changes from 4/3 to 2/3 for this example. Extensive use of the default MCCs available with the Twiddler could offer even faster typing

rates than those observed in our study. We test the effect of MCCs on expert typing rates in Section 4.3.1.

As we have shown, the same 3X4 keypad can produce vastly different typing rates. This might be due to a tradeoff in the use of space versus time. In the standard QWERTY design, all lowercase characters are devoted to a separate key (dedicated space). The opposite extreme would be to use one key to cycle through all characters one at a time. The Twiddler chording method and multi–tap are two distinct points in this design domain. Multi–tap spreads 3 or 4 letters across the keys. The user selects a letter by pressing a particular key several times. Chording does not utilize a temporal approach. The user presses multiple keys at approximately the same time to generate characters. So even if chording and multi–tap had the same keystrokes per character values, chording would be faster since it is not dependent on time.

3.5.3 QWERTY as a Baseline Predictor

We utilized the data collected from a full sized desktop QWERTY keyboard to normalize each participant's entry rate. Table 2 shows each participant's QWERTY average wpm and the ratio of his or her chording and multi–tap rates during the last session to his or her QWERTY rate. This table shows some consistency across participants despite the large range in QWERTY speeds. After twenty sessions, the average ratio for chording is 32.5% ($SD = 3.9$), while the average ratio for multi–tap is 24.7% ($SD = 4.5$).

Table 2: Typing rates as a function of QWERTY speed.

QWERTY wpm	Chording (%)	Multi–tap (%)
113.9	32.3	23.0
111.1	28.0	21.0
94.8	31.9	23.3
86.8	30.0	22.4
83.5	33.8	25.8
82.3	29.3	23.8
74.5	29.9	17.6
61.5	36.6	29.9
58.5	31.4	27.2
54.1	41.3	33.3

The consistency between participants suggests that QWERTY rates might predict chording and multi–tap rates. If someone types 90 WPM on a standard QWERTY keyboard, our data suggests that after 20 20-minute sessions she would type approximately 29 wpm chording and only 22 wpm with multi–tap.

3.6 Conclusions

In this chapter, we presented a longitudinal study comparing multi–tap and chording methods on the Twiddler, a mobile one–handed keyboard with a keypad layout similar to a mobile phone. Chording out–performs multi–tap typing speeds, is learned quickly, and our data indicates it has a higher attainable maximum rate. In addition, the chording rates reported here are faster than those reported in studies on T9 and LetterWise for similar levels of expertise. In the next chapter we present a follow up study designed to confirm the predictions of our regression curves for expert rates and explore other characteristics of expert Twiddler usage.

CHAPTER 4

EXPERT CHORDING ON THE TWIDDLER

In this chapter, we extend our previous study to confirm the rates our regression curves predicted for expert typing. We also analyze the nature of how the participants learned to type with chords. Finally, we examine the use of multi–character chords (MCCs) by our now expert typists and the effects of limited visual feedback on performance.

4.1 Towards Expertise

The first phase of this study is designed to confirm the prediction of expert rates from our previous experiment. We continued with a very similar procedure and five of our original ten subjects agreed to participate. The five that declined participation did so because of the large additional time commitment. The procedure was modified to focus on chording; we replaced the multi–tap condition from our original experiment with a second chording session. For this experiment, we compensated each participant at the rate of \$0.33 $\times$ wpm $\times$ accuracy.

We collected data for approximately 20 additional sessions resulting in a total of 40 sessions or about 13 hours of practice per participant. We ended this phase when our participants showed signs of expertise indicated by reduced rates of learning. Figure 21 shows the average typing speed across participants. Also plotted is the original regression from our first study and a modified regression based on the new data from our five participants. The dip in the typing rate at session 20 is the effect of the two week break between our original study and this follow up. While there was a decrease, the participants rebounded by the next session.

$$\text{Original regression}: \quad y = 4.8987x^{0.5781}, R^2 = 0.9849$$

$$\text{Modified regression}: \quad y = 5.3503x^{0.5280}, R^2 = 0.9787$$

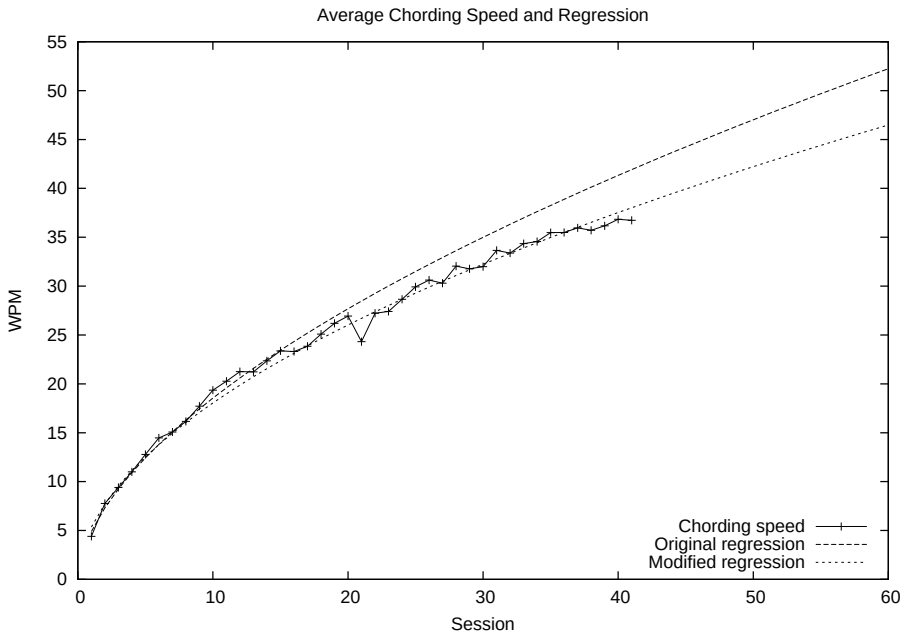

Figure 21: Mean learning rates and regression curves across participants.

After 40 sessions the average typing rate for our participants increased to 37.3 wpm. This data show that our original regression curve was slightly optimistic, predicting instead an average typing rate of 41.3 wpm. The difference could be a result of the variance in individual typing rates. Even though our regression fit to the mean typing rate of the participants is good, there are large differences in each individual's typing rate. Figure 22 shows the typing speeds for each of the participants by session. Also plotted are individual regression curves which have correlations of at least 0.96, indicating the data is well–fit. They predict that after 60 sessions, even the slowest participants would be able to type at 35 words per minute while the fastest would achieve rates in excess of 65 wpm.

Figure 23 shows the average error rate across participants using Soukoreff and Mackenzie's total error rate metric [56]. The final mean error is 6.2% and is slightly above other typing studies with a similar experimental design [38]. As shown, participants rapidly reduce their error rates as they initially learn to chord. As they learn to type faster, their accuracy gradually decreases. We believe this is an artifact of our experimental design as

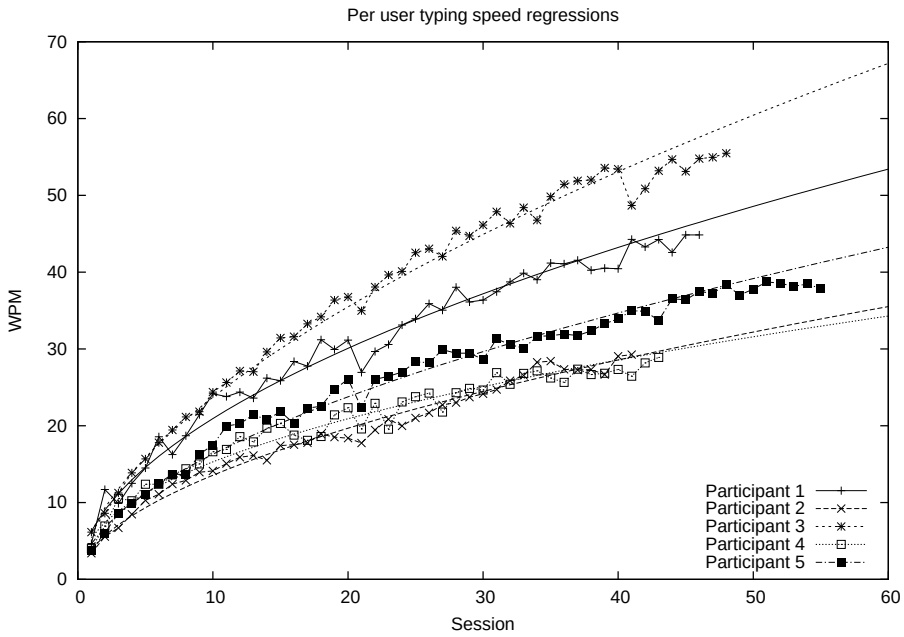

Figure 22: Per user typing rates and regressions.

we did not directly control for accuracy. Instead, each participant was compensated proportional to the product of his rate and accuracy. As a result, the participants were rewarded if a small decrease in accuracy enabled a faster typing rate. A similar effect, where error rates gradually increase as participants become experts, was shown by Matias *et al.* with the Half–QWERTY keyboard [41].

4.2 Analysis of Learning Rates

In addition to confirming the learning rate for the Twiddler, our additional data allow us to examine how users type on the Twiddler and to study the nature of the learning involved with chording. With a traditional keyboard, a character is generated by pressing and releasing a single key. Chord typing, however, may involve pressing and releasing two or more buttons to generate a character. We instrumented our experimental software to record the time each button is pressed and released for every chord. By examining the time intervals between each button press and release, we can gain insight into how novice users spend most of their time while learning and what optimizations we might make to aid performance.

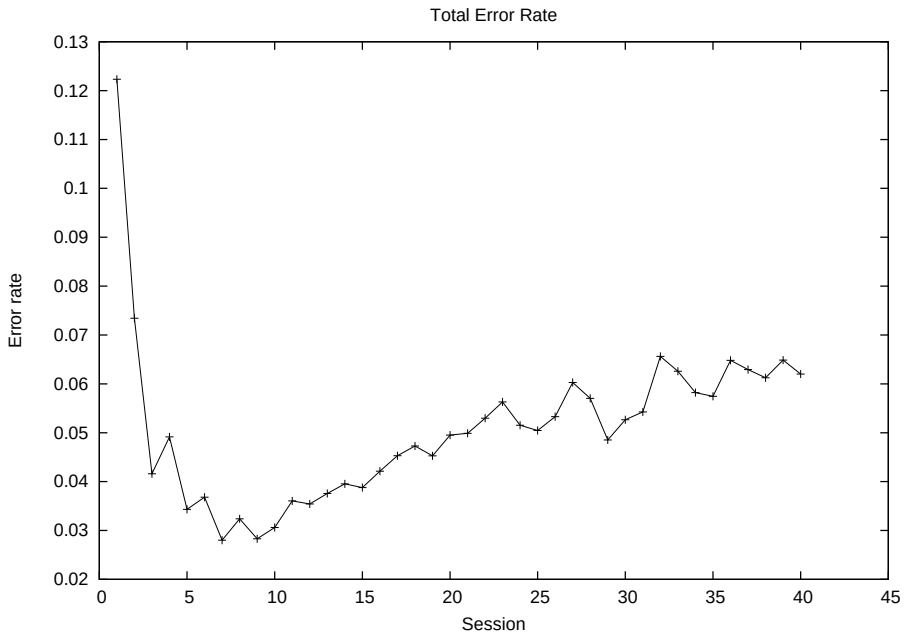

Figure 23: Mean error rate across participants.

Typing a degenerate chord involving only a single button has one press and one release. This keypress has two intervals associated with it, in–air and hold. The first interval, in–air, is the time from when the last chord was completed (all of the buttons were released) to when the button for the current chord is depressed; in other words, the time when no keys are held down. The other interval is the hold time and represents the interval between the press of the button and its release. We extended this notion of intervals to two button chords as well. The interval during which no buttons are pressed down is the in–air time, and the time during which all of the buttons are depressed is the hold time. However, the buttons in the chord may not be pressed or released at the exactly the same moment in time. This introduces two additional intervals. The time between the press of the first and second buttons of a chord is the press interval while the time between releasing the first and second buttons of a chord is the release interval. Thus, the sequence of two button chord time intervals is in–air, press, hold, and release, whereas single buttons only have in–air and hold intervals.

Figure 24 shows per–session averages of these intervals for a representative participant. This graph highlights where subjects spend their time in chording and suggests where the

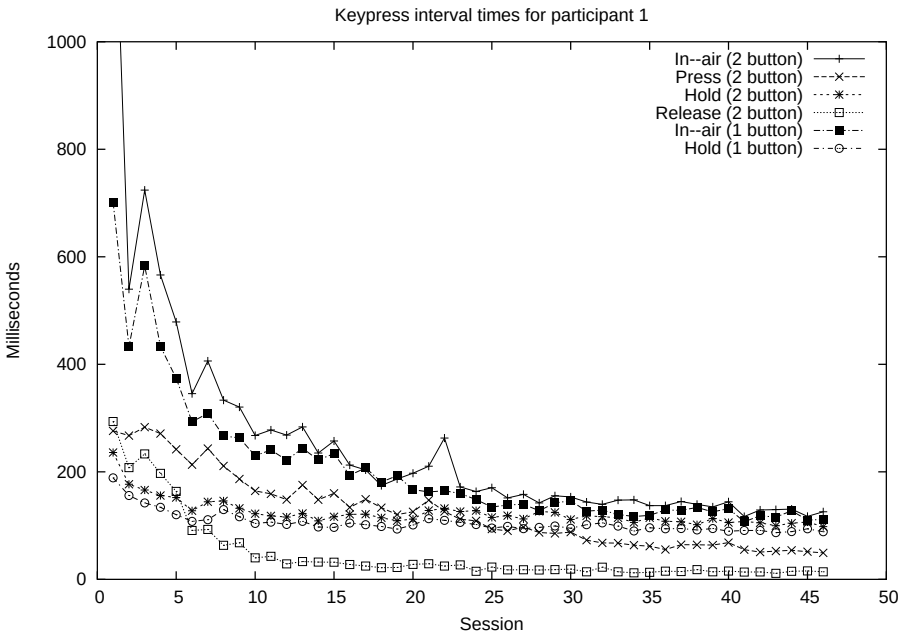

Figure 24: Keypress interval times for a single participant.

improvements of learning have the most effect. These values were computed by taking the intervals for each chord typed in sentences without any errors and then averaged for the whole session on a per user basis. We did not include sentences with errors as we did not want to confound our data. Mistyping one chord can impact several others, and it is not straightforward to incorporate the error data with our individual time intervals.

4.2.1 In–air Interval

All of the participants' average in–air intervals for single and two button chords are shown in Figure 25 and Figure 26 respectively. These time intervals exhibit the largest effects of learning. For novices, it is likely that this interval is dominated by the cognitive effort associated with remembering how to type each character and how to move their fingers to the correct position to type the letter. For experts, the delay becomes dominated by the time it takes to move the fingers from one chord to another. Comparing the in–air interval for single and two button chords reveals that, on a per user basis, the single button times are slightly faster and show better rates of learning. However, the two button in–air interval tracks the single button interval rather well. By the end of the study, the difference between

the times on a per user basis becomes much smaller. On average our participants use 244ms to type a single button chord and 354ms for a two button chord. The discrepancy is mostly due to a single participant (number two) who lagged behind on learning the two button chords. With additional practice his rates would approach the others, and the difference between the in-air times for single and two button chords would decrease.

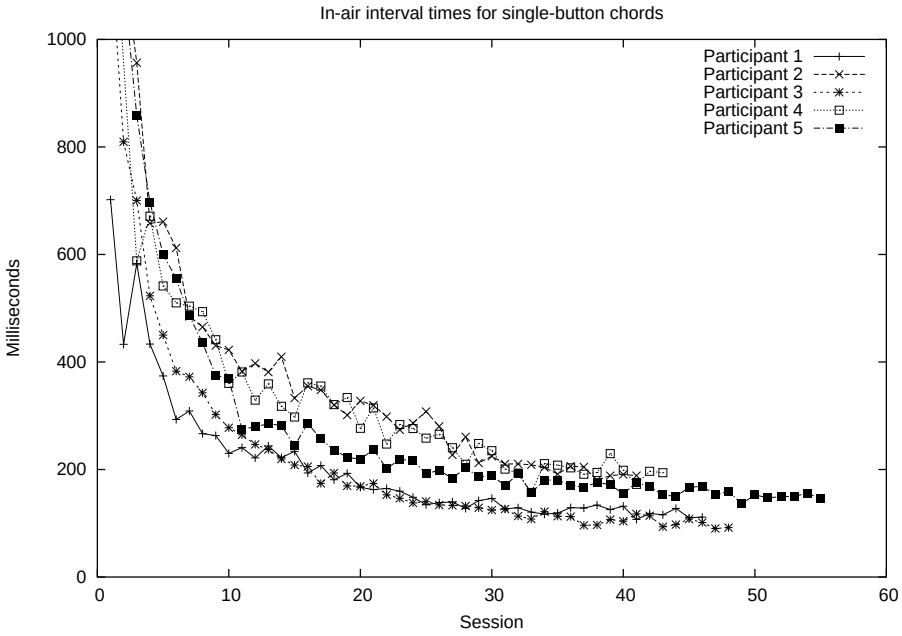

Figure 25: In–air interval times for single button chords.

4.2.2 Press Interval

Figure 27 presents the press interval, which is the time between the first and second buttons of a chord being pressed. This interval is particularly interesting because it reveals different typing strategies between users. A single participant (number 3) always pushes both of the buttons in a chord at nearly the exact same time. The average delay between the first and second button press is only 7.25ms indicating that he always presses both buttons as one action. The other participants show a larger delay between these button presses, indicating that they press the buttons sequentially and likely learned how to press the chords in a different way than participant 3. The delay could be from planning and executing the two button presses in the chord separately. The slower users may also initially wait for

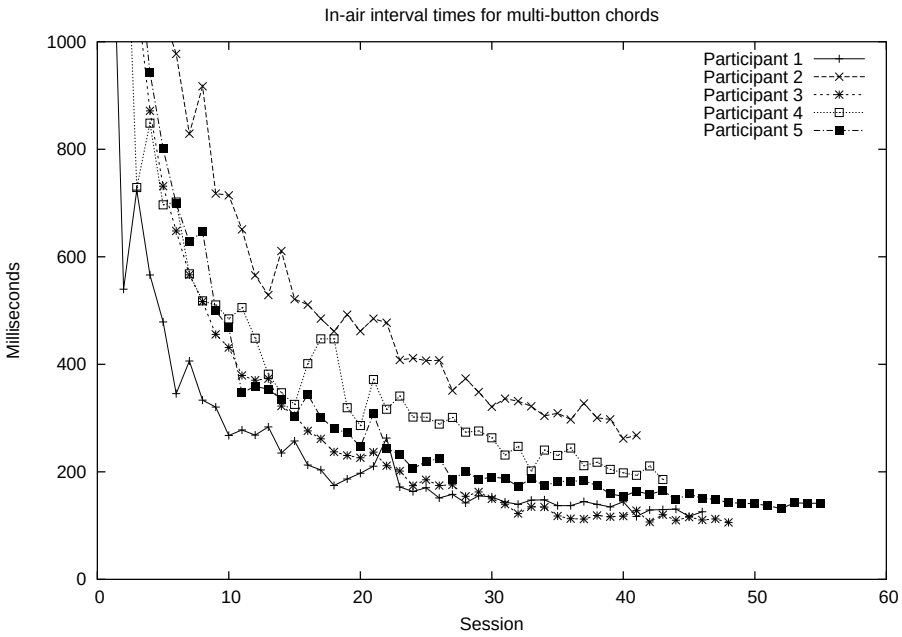

Figure 26: In–air interval times for two button chords.

haptic feedback after pressing the first button. For these participants there is some learning associated with this interval; however, the in–air interval is more pronounced.

This interval may also have implications for expert typing rates. Participant 3 was significantly faster than the other participants and was typing at 67 wpm by the conclusion of our experiments. To determine if this might be attributable to his simultaneous press strategy, we examined the data from the other five participants from our original study who had stopped after 20 sessions. Two of the subjects employed the simultaneous press strategy, two of them the sequential strategy, and one started out sequential but appeared to switch mostly over to the simultaneous strategy by the end of the twenty sessions. The participants who used the simultaneous press strategy were no faster than those who used the sequential strategy. While simultaneous pressing might not produce the fastest rates while learning, it should be very beneficial to experts. At 60 words per minute, the average time to type one character is 200ms. Since the press interval times varied up to 100ms by the end of this phase and apply to more than 66% of the alphabet, pressing both buttons of a chord at the same time should increase the typing rate.

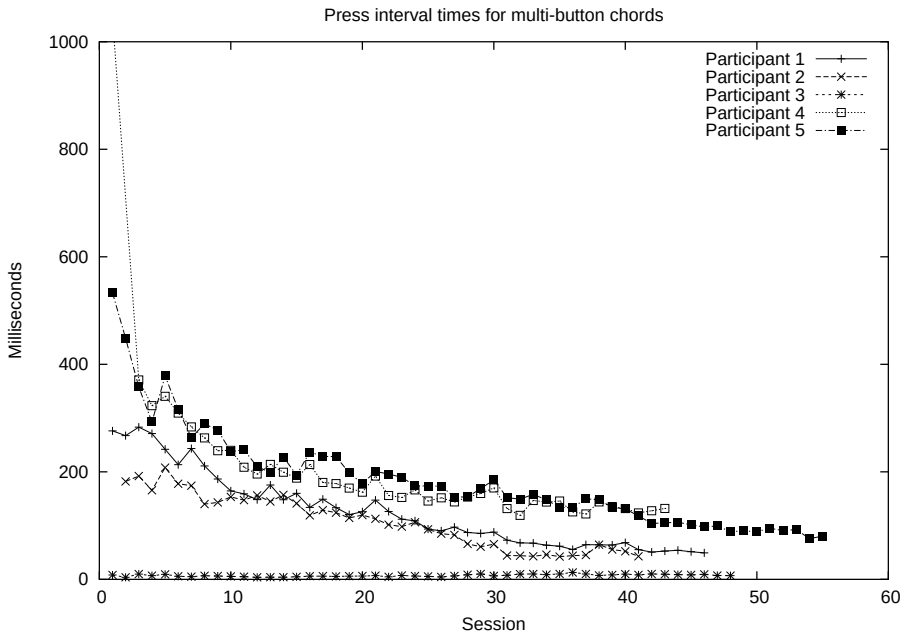

Figure 27: Press interval times (two–button chords).

4.2.3 Hold and Release Intervals

Our last two time intervals are the hold interval (Figures 28 and 29) and the release interval (Figure 30). The average hold interval shows slight improvement with practice, and in general single button chords are held for slightly less time. At the end of this phase of the experiment, the single button chords were held 98ms while two button chords were held 107ms. Perhaps participants spent the extra time to ensure that they avoid releasing the first finger before the second one is depressed. Finally, while only one participant pressed both keys of a chord simultaneously all of the participants rapidly learned to release both buttons of a chord at approximately the same time. After about 10 sessions most of the users released both keys in less than 25ms.

4.3 Expert Usage

After approximately 45 sessions, enough data had been collected that we could be confident of our regressions' predictions. While performance was still improving, the rate of learning had decreased enough that we considered our participants to be expert users. At this point we continued our experiment with two additional phases designed to investigate various

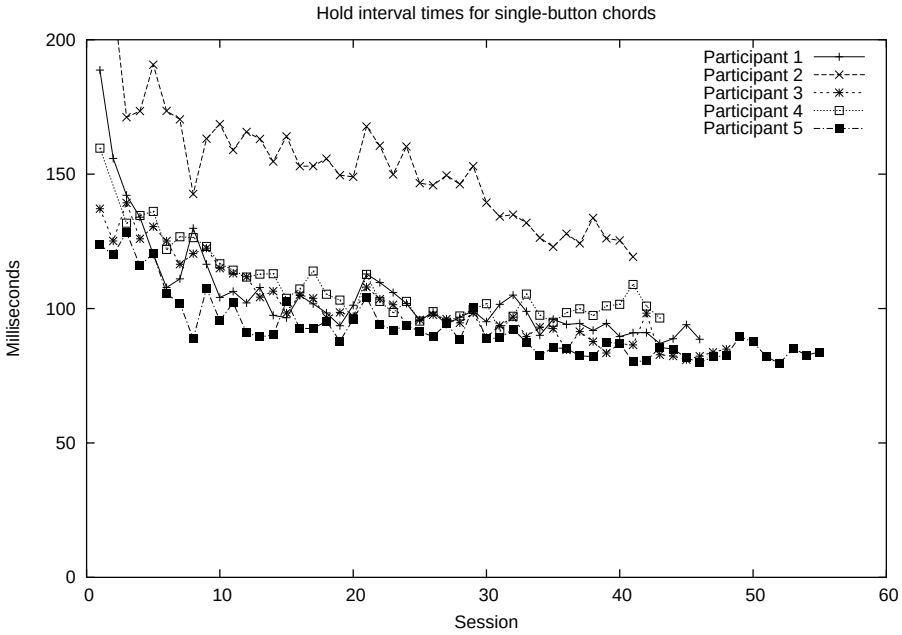

Figure 28: Hold intervals for single button chords.

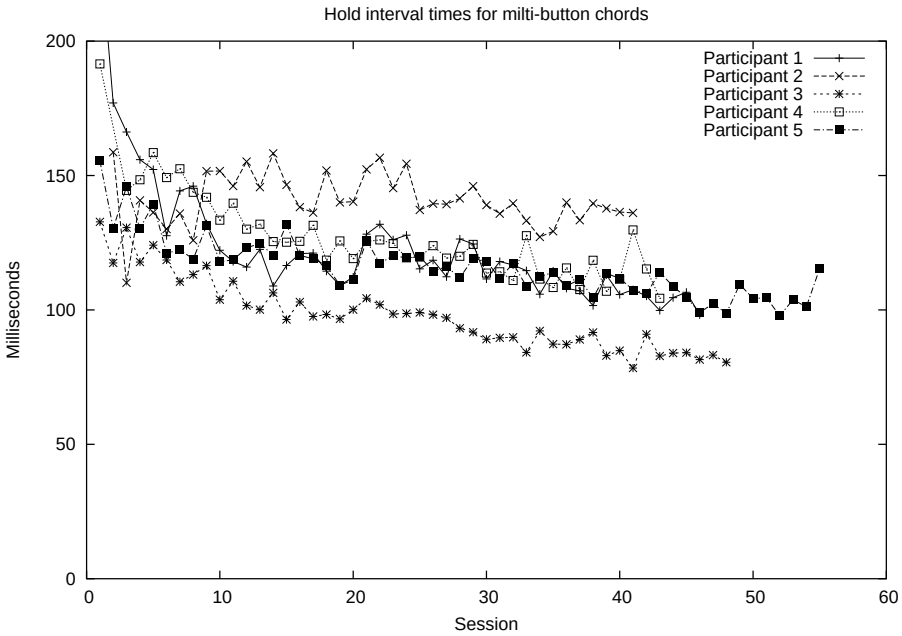

Figure 29: Hold intervals for two button chords.

aspects of expert typing. We examined the possible benefits of multi–character chords (MCCs) and the effects of typing with reduced visual feedback (blind typing).

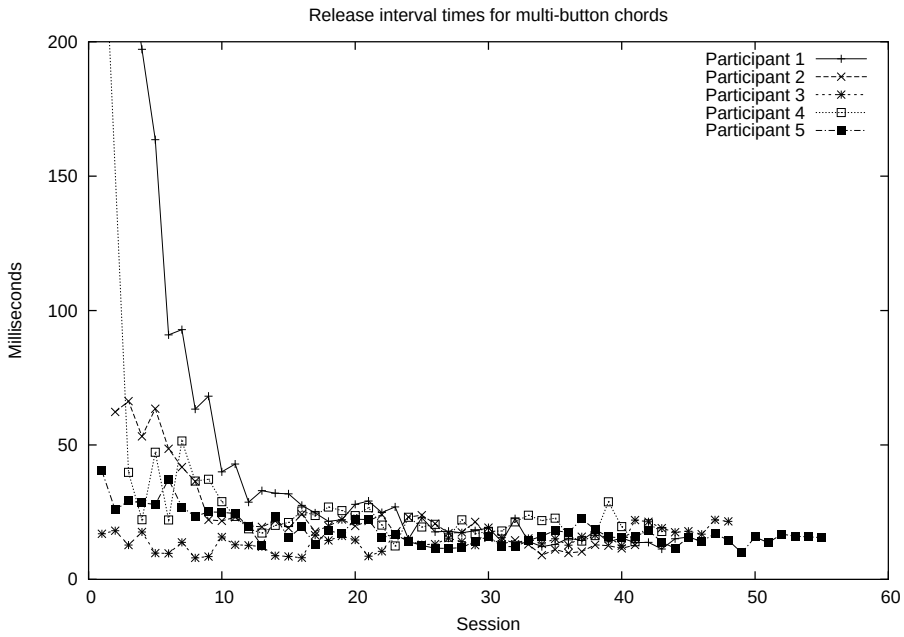

Figure 30: Release interval times (two–button chords).

4.3.1 Multi–Character Chords

As mentioned previously, there are 255 possible chords that can be typed on the Twiddler using the four fingers. Of these, only a small subset are allocated to the alphabet and punctuation needed to type English text. Some of the unused chords can be employed as multi–character chords (MCCs) which could generate any sequence of characters. In the next phase of our experiment we wanted to determine if MCCs for short common words and suffixes would improve our participants' typing rates. Our hypothesis was that MCCs would have a positive impact on typing rate because the number of button presses needed to type any given MCC string, such as "the ", would be reduced to one chord. Using a MCC would reduce the overall number of keystrokes per character (KSPC) [37] as fewer keystrokes (button presses) would be needed to generate the same text.

Using word frequency data from the commonly used text corpus, the British National Corpus [30], we selected 12 strings of at least three letters that are prevalent in written English. For this experiment we selected 'for', 'and', 'the', 'ent', 'ing', 'tion', 'ter', 'was', 'that', 'his', 'all', and 'you' to be typed as MCCs. We assigned these strings to unused chords that did not involve the index finger. As many of these strings are normally followed

Table 3: Keymap for new multi-character chords (MCCs) with and without trailing space.

Buttons	String	Buttons	String
0LL0	'for'	RLL0	'for '
0MM0	'and'	RMM0	'and '
0RR0	'the'	RRR0	'the '
00LL	'ent'	R0LL	'ent '
00MM	'ing'	R0MM	'ing '
00RR	'tion'	R0RR	'tion '
0LLL	'ter'	RLLL	'ter '
0MMM	'was'	RMMM	'was '
0RRR	'that'	RRRR	'that '
0L0L	'his'	RL0L	'his '
0M0M	'all'	RM0M	'all '
0R0R	'you'	RR0R	'you '

by a space character, this assignment enabled us to add 12 extra MCCs that had a trailing space such as 'the '. The buttons used for these chords are the same as the normal version, only the user also depresses the button used for space (the right button operated by the index finger). Table 3 shows the keymap for the additional MCCs.

Our experimental software has a diagram of the Twiddler keypad that was designed to act as a guide to help the users learn the basic alphabet keymap. We modified the diagram so that the keys needed for the MCC are also highlighted (Figure 31). To encourage the use of MCCs, we modified the error calculation so that typing the MCC string letter–by–letter counted against the participant's accuracy.

The effect of MCCs on our participants' typing rates is mixed. Initially, our participants typed more slowly when using MCCs as they were novices for those chords. For the first session, the average typing speed dropped to 83.5% of what it had been. However on the fifth session, the average speed was 97.1% of the pre–MCC speed, and by the tenth session it was 104.5% and continued to improve. Even though the rate increased beyond the typing speed just before the introduction of MCCs, the participants were still slowly learning. If we had not introduced MCCs and just had our participants continue to practice, we would have expected the rate to increase to approximately 112% based upon our regressions. As a result we cannot attribute the overall increase in typing rate solely to the effects of MCCs.

To better understand the effects of MCCs, we compared the amount of time participants

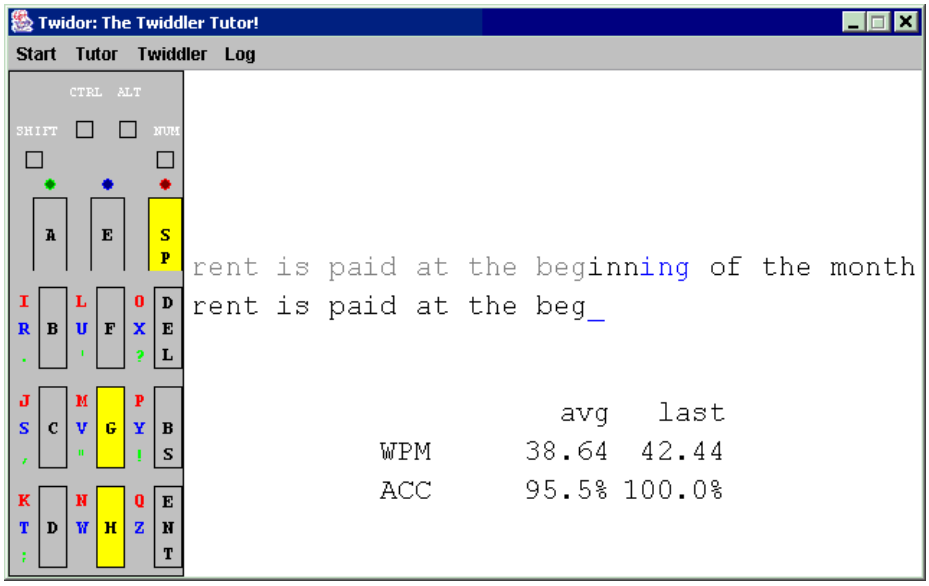

Figure 31: Our experimental software showing the use of MCCs; "ing " is the MCC to be typed ('R0MM') and is highlighted in blue.

needed to type the MCC strings letter–by–letter just before the introduction of MCCs and the time needed to type the new chord. On average, participants typed the MCC strings using the multi–character chord in 58.5% of the time it took to type the same characters letter–by–letter (596ms vs 1018ms).

An analysis of our phrase set revealed that 17.5% of the characters in our phrase set can be typed with MCCs. Weighted by the frequency of MCCs in our phrase set, this would correspond to approximately an 8% increase in average overall typing speed. This effect would likely be more pronounced using a phrase set more representative of English on a word frequency basis instead of letter frequency [39] as our participants master the new multi–character chords.

At the end of the MCC phase, our participants required an average of 596ms to type each multi–character chord and were still showing signs of improvement with MCCs. While our multi–character chords might be slower in general because they involve up to four buttons, the chords for the alphabet that require two buttons only take 354 ms on average which is only 31.3% more time than typing a single button chord. As a result, we expect MCC rates would improve once our participants mastered typing the multi–character chords.

4.3.2 Blind Typing

In a mobile environment, a user's visual attention may be diverted away from her display while entering text. For instance, with our case study of an expert wearable user (Chapter 2), our participant indicated that he would maintain eye contact with his conversational partner. This practice limits the effectiveness of visual feedback while typing. Silfverberg examined the effect of visual and tactile feedback when using a mobile phone keypad [53]. He found that limited visual feedback combined with low tactile feedback hinders a user's average error rate; on the other hand, good tactile feedback results in a much smaller increase in errors.

Inspired by Silfverberg's study, our expert case study and our own anecdotal experience of typing with limited visual feedback, we designed the last phase of this experiment to evaluate blind typing on the Twiddler. We designed 3 conditions (normal feedback, dots feedback, and blind) over 5 sessions of typing. Each condition took 15 minutes. Our normal feedback condition displayed the text typed under the phrase presented to the participant as shown in Figure 31 but without highlighting. As the Twiddler is held with the keypad facing away from the user, this condition corresponds most closely to Silfverberg's indirect visual feedback condition. For our dots condition, we displayed periods for each character typed instead of the transcribed text. Thus, participants saw their position in the supplied phrase, but not specifically what they typed. This condition is designed to simulate monitoring text typed without being able to actually read the letters such as seeing the text on a heads–up display using only peripheral vision. Finally, the blind condition does not show any on–screen indication of what is typed and mimics Silfverberg's no visual feedback condition. For both the dots and blind conditions, participants were shown their transcribed text and error statistics when they pressed enter at the end of the phrase. We predicted that, like Silfverberg, reducing the visual feedback would limit our participants' typing rate and accuracy.

Surprisingly, changing the visual feedback did not hinder the participants in their typing as expected. In some cases typing rates and error improved with the reduced feedback. Table 4 shows the change in speeds and the error rates for the typing conditions. Values

Table 4: Per participant typing and error rates for the three conditions. Bold indicates a statistically significant difference at the 0.05 level between that condition and the normal condition for that user.

Typing Rates (wpm)					
Participant	1	2	3	4	5
Normal	51.8	37.6	64.2	36.2	41.8
Dots	51.7	37.5	**67.2**	36.0	**43.1**
Blind	**53.7**	37.5	**67.7**	36.6	41.7

Percent Errors					
Participant	1	2	3	4	5
Normal	5.61	5.62	7.01	9.83	6.64
Dots	**4.82**	5.02	**5.75**	9.26	5.83
Blind	5.03	**4.63**	**5.90**	8.89	**5.44**

where a two–tailed t–test showed a statistically significant difference at the 0.05 level from the normal condition are marked with bold. Whenever there is a statistically significant difference between normal typing and one of the reduced feedback conditions, the reduced feedback condition shows an improved typing rate or a reduced error rate. One possible explanation for this effect is that subjects are operating with open–loop motor control in the blind conditions. When there is visual feedback, the user switches to a closed–loop mode and incorporates the visual feedback into her typing process, thus requiring slightly more time.

4.3.3 Expert Typing Rates

By the end of all of our experiments, our participants completed an average of 75 sessions which corresponds to approximately 25 total hours of practice. Figure 32 shows the typing rates for our participants across all of our experimental conditions described above. The final average typing rate reached 47 wpm and unexpectedly our fastest participant achieved a rate of 67.1 wpm which is as fast as an expert which has been using the Twiddler for ten years.

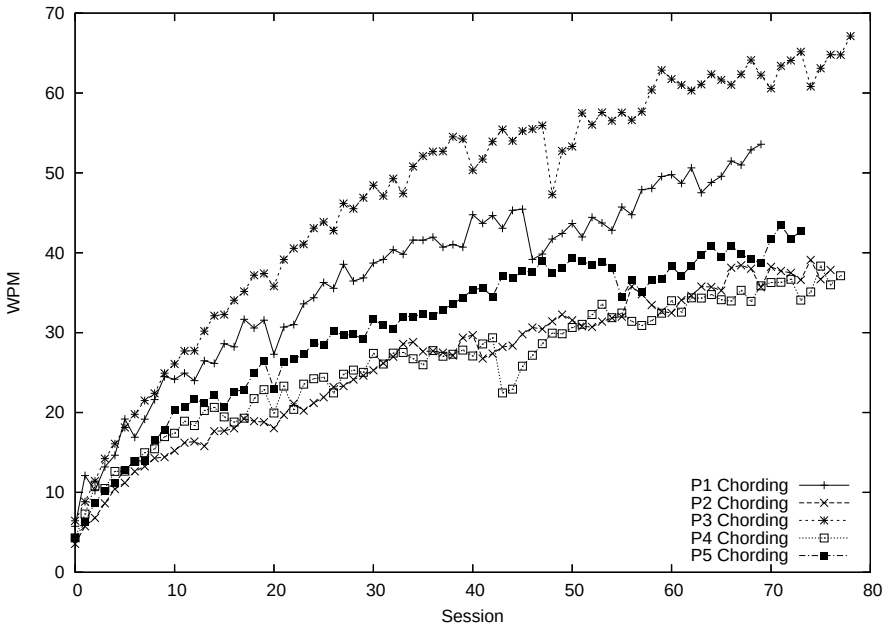

Figure 32: Data across all phases of experiment for all 5 participants.

4.4 Conclusions

We have analyzed various aspects of expert chording on the Twiddler keyboard including text entry speed, the effects of visual feedback, and the use of multi–character chords. We found that our participants reached an average typing rate of 47 wpm while our fastest participant reached 67 wpm. Our data on multi–character chords indicated that they could provide even higher typing rates. We examined how our participants learned to chord, showing most of the speed increase associated with learning occurs during the in–air time interval. We also found a difference in strategy of how our participants press the buttons of a chord. The blind typing data shows that the Twiddler can be used effectively with limited visual feedback which is important in a mobile environment. Given the expert users' high text entry speeds and ability to touch type, chording seems to be a viable mechanism for text entry on future mobile devices.

The experiments from this chapter and the previous show the great potential of the Twiddler keyboard. It offers a very rapid learning rate and a high maximum typing rate. Our work on blind typing also suggests one reason the Twiddler could be so successful for use in everyday conversational situations as seen in the case study from Chapter 2. In the next chapter we explore enhancing novice performance with the Twiddler.

CHAPTER 5

ENHANCING NOVICE TWIDDLER USE

Our final work on the Twiddler is focused on the novice user. In this chapter we present a study examining the effects a chording tutor might have on novice typing rates. While the Twiddler shows great potential for permitting rapid text entry in a mobile environment, our studies did show that the initial typing rate was about half that of multi–tap (Chapter 3). In this chapter, we explore ways to improve novice typing rates and acceptance of the Twiddler.

5.1 Aiding Novice Twiddler Typing

The orientation of the hand while typing on the Twiddler is more like a musical instrument such as a guitar than a computer keyboard. While offering good expert rates, this orientation poses a problem for novices; it makes "hunt–and–peck" typing difficult. To look at which key to press, a user must to rotate the Twiddler out of typing position to bring the keypad into view. The second potential barrier for novice users is chording, pressing multiple buttons simultaneously to generate a character. While chording is employed on desktop keyboards (shift, control and alt are often used as one button of a chord) it is more rare on mobile phone keypads. Furthermore for the Twiddler, the majority of the characters in the alphabet require the use of chording. To address these potential problems, we explore two aids that might aid novice users: a structured phrase set and software highlighting for the keys to be pressed.

5.1.1 Phrase Set

Our first aid employs a phrase set tailored to the Twiddler keymap. One common practice with tutors for desktop keyboards is to subdivide the alphabet based on the physical layout of the keyboard. For instance, the software starts by teaching the user the "home row" and gradually adds more letters to be learned based on the position of the keys on the keyboard.

We extend this analogy to the Twiddler keymap and different phrases that exercise different categories of chords. Our new phrase set is initially restricted so that the user only types letters requiring a single button press (‘a’–‘h’). Next the phrase set is changed so the participant types just the chords that involve the red shift (‘i’–‘q’). Then, the phrase set uses the combination of single and red (‘a’-‘q’), followed by just blue (‘r’-‘z’), single and blue, and finally all of the letters.

In addition, empirical evidence from psychology studies indicates that simplifying a complex task into smaller tasks can reduce the workload associated with learning the complex task and can reduce error rates [11, 28]. Our new phrase set can be ordered so that the task of learning all 26 letters of the alphabet is simplified into several subtasks. Each task focuses on learning subsets of the alphabet where each subset is associated with a critical gross physical movement. By segmenting the phrase set based on the different types of chords, we can help the user focus on the different types of physical movements needed to type. The phrases that use only a single button let the user explore the keyboard. The red and blue phrases give practice for the motions needed to type the different chords involving the two shift keys. Finally, the phrases which use combinations transition the user to more realistic text and the associated movements required.

5.1.2 Highlighting

Our second aid supplements an on–screen keyboard representation which provides the user with a reference of the mapping between buttons and characters (Figures 33 and 34). The representation is shown to the user on the left–hand portion of the display (Figure 16) and is the same as the representation which is printed on the faceplate of the Twiddler. All of the characters for single button chords are printed on the button. The characters for the rest of the chords are printed in their respective colors next to the appropriate button.

We provided this on–screen representation in our previous studies so that our participants could use it as a reference while learning to type. It is designed to help reduce the need to turn the Twiddler in order to look at the keypad. Instead, participants can scan the on–screen representation to find the letter they need to type. Once the correct letter

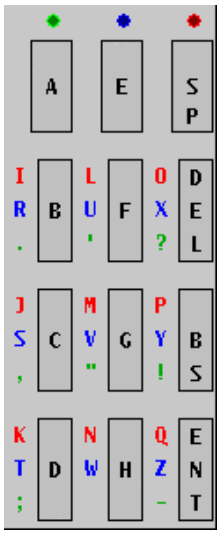

Figure 33: Graphical representation of Twiddler chording keymap shown with highlighting off.

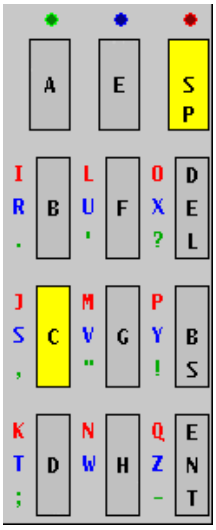

Figure 34: Graphical representation of Twiddler chording keymap shown with highlighting on.

is found, the participants can determine which buttons to press. While informative, it is visually busy and requires some experience to understand and use. To facilitate the use of the Twiddler representation, our software can highlight the next set of buttons the user is to press (Figure 34). The highlighting is designed to reduce the amount of time the user spends visually scanning the representation. When highlighting is turned on, the buttons to be pressed for the next character change color.

We present our study designed to explore these two aids. Our goal is to determine if either aid can improve novice Twiddler typing and see what combinations can lead to the best novice typing rates and least workload.

5.2 Experiment: Comparing Novice Aids

Our experiment retains the same core design from our previous Twiddler studies (Chapter 3 and 4) which were based on other text entry research [38, 40]. For these studies, experimental software presents a sequence of phrases one at a time and the participants are asked to type the displayed text. Phrases are grouped into twenty minute typing sessions and the experimental variables can be manipulated per session.

Table 5: Example phrases exercising different portions of the Twiddler keymap.

Characters	Example Phrase
Single	dad added a facade
Red	i look ill in pink
Single + Red	a feminine chief in old age
Blue	suzy trusts wussy russ
Single + Blue	the greatest war there ever was

5.2.1 Design

We are using two twenty minute sessions in this experiment: practice and evaluation. For the practice session, we manipulate our experimental variables across participants while the evaluation session is the same for all participants.

Our first variable tested in the practice session corresponds to the phrase set aid. Our Twiddler phrase set has 14 phrases that require only single button presses, 14 phrases that only require the red shift and 14 for blue. We have 26 phrases that use single and red characters and 25 that use single plus blue. Table 5 shows some example phrases from each of our categories. In total, our 93 phrases have an average length of approximately 25 characters and the correlation with the frequency of characters in English is 89% [39]. Using these phrases, we have two conditions. The first condition, "ordered," presents the phrases in a structured order. Initially, our software randomly selects phrases that require single button presses. Next it uses only "red" phrases, then single plus red, blue, and single plus blue. Our second condition, "unordered," randomly displays any of the phrases for the whole period. This condition allows us to control the content of phrase set but does not offer the aid of learning in sequence. For our evaluation session, we use the phrase set developed by MacKenzie and Soukoreff [39]. These phrases average approximately 28 characters each and are selected randomly from the set of 500 total phrases. The phrases contain only letters and spaces, and we altered the phrases to use only lower case and American English spellings. These are phrases specifically designed as representative samples of the English language and have a correlation with English of 95%.

Our second variable is highlighting, and we are testing three highlighting modes during

the practice session: no highlight, always on highlight, and delayed highlight. For no highlighting, the on–screen representation is shown but does not change. When highlighting is turned on, the buttons for the next character to be typed are highlighted in yellow (Figure 34). Our software also has a delayed highlighting option, in which initially no buttons are highlighted. In this case, no prompt is shown initially. If there is no activity, the keys to press are highlighted following a delay. After pilot testing a 1.5s delay was chosen. This value was large enough to allow the pilot subjects to type many of the characters they had already learned without the highlight appearing. This value also corresponds to typing at 8 wpm which, as discussed previously, is the rate at which many novices type with other mobile phone methods. For the practice session, each participant is assigned to one of the three highlighting categories. For the evaluation session, highlighting is turned off for all participants. As a result, our experiment is a 3 x 2 design. We have three highlighting and two phrase set possibilities resulting in a total of six between–subject conditions.

5.2.2 Participants

We recruited 60 students from the Institute. The majority participated in return for credit in their respective courses and a few students volunteered. As in our previous experiments, all of our participants had no experience with the Twiddler. Each participant was assigned randomly to one of the six conditions resulting in ten participants per condition. Our participants ranged in age from 18 to 37 years old and had a mean age of 20.9 ($SD = 3.7$). Thirty–one participants were female and four left handed. Twelve participants were non–native English speakers. The non–native speakers had been speaking English on average 8.9 years ($SD = 6.4$). Fifty–two of our participants were mobile phone owners. The owners made an average of 6.6 calls per day ($SD = 5.4$) and sent an average of 2.3 text messages each day ($SD = 4.5$).

5.2.3 Procedure

The experiment takes approximately 90 minutes to complete. It begins with the researcher presenting an overview of the experiment, and consent and demographic forms are filled out.

Next, each participant types using a standard desktop QWERTY keyboard for three minutes. We collected this data to obtain a baseline typing rate for each participant. Following the desktop keyboard test, the participants are given written instructions explaining how to hold and type with the Twiddler and how the typing software works. As appropriate, the instructions explain the breakdown of the phrase set and how the highlighting works. For each segment of the study, we instruct the participants to type "as quickly and accurately as possible."

The first session of Twiddler typing starts next. The practice session begins with a warm–up round which consists of typing the two phrases, "abcd efgh ijkl" "mnop qrst uvwx yz" twice. The warm–up data is not used in measuring performance. After the warm–up, the participant begins the practice session. At this point the twenty minute timer starts and data recording begins. The practice session is divided into six blocks. If the participant is using the ordered phrase set, each block switches from one set of chords to the next. Four minutes is spent on single phrases, and four on red. This is followed by two minutes of practice using the single plus red phrases. Next is four minutes of blue and two minutes of single plus blue. Finally, there is four minutes of typing where phrases are selected randomly from the entire phrase set resulting in twenty minutes total. The unordered condition uses blocks of the same duration; however for each block, phrases are selected randomly from the entire phrase set. Once the twenty minutes of the practice session are over, the participants take a five minute typing break. During the break they fill out a NASA Task Load Index (TLX) questionnaire [20]. The evaluation session starts once the questionnaire is completed and the break is over.

At the beginning of the evaluation session the participants are instructed that the highlighting will be turned off for the upcoming session (for those who had highlighting in the practice session). At this point, the software switches to using the MacKenzie phrase set for all participants. After typing the alphabet twice, participants resume the experiment. The evaluation session is divided into four blocks of five minutes. At the end of the twenty minute session, participants fill out a second NASA-TLX questionnaire based on the evaluation session only.

5.2.4 Software and Equipment

As in our previous experiments, the testing software is self–administered under researcher supervision. It presents the participants with the key layout for chording (Figure 33) and statistics of performance so participants can monitor their progress. A phrase is displayed on the screen, and the subject's typed text appears immediately below the presented text (Figure 16). The software was modified to include a built–in scripting engine used to configure and control the experimental conditions. We have six scripts (one for each of our conditions) that are used by the software to run the participants through our procedure.

5.3 Results

Across our 60 participants we collected approximately 3500 phrases of chording data which resulted in 84,000 transcribed characters. Using this data, we examine the effects of our experimental manipulations on participants' typing speed, error rate, and workload. We performed a 3 (highlighting) x 2 (phrase set) x 2 (session) ANOVA on each measure. Highlighting and phrase set are between-subject variables, while session is a within-subject variable. The inclusion of session allows us to determine the presence and magnitude of typing speed changes between the practice and evaluation sessions. Where appropriate, we also examine the individual 2–way interactions and simple effects of each manipulation. All results are interpreted using $\alpha = 0.05$.

5.3.1 Text Entry Rates

First, we examine the effect our conditions have on typing speed which is measured in words per minute (wpm). For each participant, we calculated the cumulative wpm value across an entire session by taking the sum of the total number of words and dividing by the total time spent typing in the session. Table 6 displays each group's mean wpm and standard deviation for both the practice and evaluation sessions.

There is no 3–way interaction between our variables, $F_{(2,54)} = 1.12, p = 0.34$, $MSE = 1.00$, but there is a significant interaction between the highlight manipulation and session, $F_{(2,54)} = 8.43, p = 0.001, MSE = 7.58$. A simple effects analysis demonstrates that

Table 6: Mean typing rates in words per minute (with standard deviations) for the practice and evaluation sessions for all 6 groups.

Practice Session	Highlight off	Highlight delay	Highlight on	*Mean*
ordered	6.61 (2.58)	6.73 (1.20)	6.21 (1.16)	6.52 (1.72)
unordered	5.17 (1.74)	4.88 (1.15)	6.34 (1.69)	5.46 (1.63)
Mean	5.89 (2.26)	5.81 (1.49)	6.28 (1.41)	5.99 (1.75)

Evaluation Session	Highlight off	Highlight delay	Highlight on	*Mean*
ordered	6.92 (2.15)	6.61 (1.50)	5.42 (1.80)	6.32 (1.89)
unordered	6.69 (1.87)	5.69 (1.48)	5.55 (2.20)	5.98 (1.88)
Mean	6.80 (1.96)	6.15 (1.53)	5.48 (1.96)	6.15 (1.88)

the highlighting off group typed slower in the practice session than in the evaluation session, $F_{(1,54)} = 9.32, p < 0.01$. In contrast, the highlighting on group typed faster in the practice session than in the evaluation session, $F_{(1,54)} = 7.02, p = 0.01$. The delay group exhibited no reliable difference in typing rate between the practice and evaluation sessions, $F_{(1,54)} = 1.33, p = 0.25$. A simple effects analysis of highlighting for each session revealed no significant differences, suggesting that there is no overall difference between the highlighting groups for the practice session, nor for the evaluation session.

There is a significant interaction between phrase set and session, $F_{(1,54)} = 4.26, p = 0.04, MSE = 3.83$. Simple effects analysis of phrase set in the practice session reveals that the ordered phrase set group typed faster than the unordered phrase set group, $F_{(1,54)} = 6.01, p = 0.02, MSE = 2.77$. In contrast, during the evaluation session there is no significant difference between phrase set groups, $F_{(1,54)} = 0.501, p = 0.48, MSE = 3.44$. Next, we examine if the typing rate changed between the practice and evaluation sessions for either phrase set group. A simple effects analysis reveals that the unordered group's rate increased from practice to evaluation, $F_{(1,54)} = 4.39, p = 0.04$. In contrast, the ordered group's rate

did not differ between practice and evaluation, $F_{(1,54)} = 0.68, p = 0.41$. These results suggest that during the practice session the ordered phrase set allows faster typing than the unordered phrase set. In the evaluation session, when all participants typed using the same phrase set, there is no statistical difference in the typing rate.

5.3.2 Error Rates

Next, we examine the number of errors made. Table 7 shows the percent error means and standard deviations for each group. We are using Soukoreff's and Mackenzie's total error rate metric [56]. This metric accounts for both corrected and uncorrected errors made by the participants and provides a single total error rate.

Table 7: Mean percent error (with standard deviations) for the practice and evaluation sessions per group.

Practice Session	Highlight off	Highlight delay	Highlight on	*Mean*
ordered	19.9 (11.7)	14.5 (4.8)	12.1 (5.5)	15.5 (8.4)
unordered	13.6 (6.3)	15.7 (6.7)	10.0 (4.4)	13.1 (6.1)
Mean	16.8 (9.7)	15.1 (5.7)	11.1 (4.9)	14.3 (7.4)

Evaluation Session	Highlight off	Highlight delay	Highlight on	*Mean*
ordered	15.2 (10.0)	13.0 (8.4)	15.5 (7.5)	14.6 (8.5)
unordered	13.0 (6.9)	13.1 (3.4)	13.6 (7.9)	13.2 (6.1)
Mean	14.1 (8.4)	13.0 (6.3)	14.5 (7.5)	13.9 (7.4)

There is no significant 3–way interaction, $F_{(2,54)} = 0.737, p = 0.48, MSE = 0.002$, indicating that we can analyze the data as three 2–way ANOVAs. There is also no significant interaction between phrase set and highlighting, nor between phrase set and session for the participants' error rates.

As with typing rate, there is a significant interaction between highlighting and session,

$F_{(2,54)} = 4.59, p = 0.01$, $MSE = 0.002$. Using a simple effects analysis, we can determine how the highlighting manipulation changes as a function of session. Highlighting has a significant effect on error rates in the practice session, $F_{(2,54)} = 3.50, p = 0.04, MSE = 0.005$, but not in the evaluation session, $F_{(2,54)} = 0.21, p = .82$. A post–hoc contrast reveals that in the practice session the highlighting on group made fewer errors than the other two highlighting groups, $t_{(57)} = 2.50, p = 0.02$.

Next, we examine how the highlighting manipulations impact error rates as participants move from the practice to evaluation sessions. For participants with highlighting on, error rates increase between the practice and evaluation sessions, $F_{(1,54)} = 4.85, p = 0.03$. There is no significant error rate differences between the practice and evaluation sessions for either the highlighting off group, $F_{(1,54)} = 2.88, p = 0.10$, or the delay highlighting group, $F_{(1,54)} = 1.68, p = 0.20$. This result suggests that error rates, which are significantly lower for the group with highlighting on during the practice session, increased to the level of the other highlighting groups during the evaluation session.

5.3.3 Workload

The NASA Task Load Index (TLX) questionnaire measures subjective workload ratings. Previous studies have indicated that it is a reliable and valid measure of the workload imposed by a task [20, 24]. Subjective workload ratings can be more sensitive to working memory demands than measures of performance [69]; this is important given the need for the participants to remember the Twiddler key mapping. Additionally, subjective ratings can be informative when a task is difficult yet within the individual's capability. For instance, as a task becomes more difficult, the individual can increase his or her effort in order to maintain the same level of performance. In this case, subjective ratings of workload could capture this increased effort, whereas performance measures could not [69].

The NASA-TLX consists of six scales: mental demand, physical demand, temporal demand, performance, effort, and frustration; each scale has 21 gradations. For each scale, individuals rate the demands imposed by the task. In addition, they rank each scale's contribution to the total workload by completing 15 pairwise comparisons between each

combination of scales. This procedure allows an investigation of the task demands load on each scale, as well as a measure of the global workload.

Interpretation of the mental, physical, and temporal demand scales are straightforward; each scale captures the demand imposed by its title. The performance scale captures how successful participants felt they were at accomplishing the given task. The effort scale captures how hard individuals had to work in order to achieve their level of performance; both mental and physical effort can contribute to this scale. The frustration scale captures how much the task annoys or discourages individuals.

The overall workload rating is calculated by summing the product of each scale's rating and weight. This calculation results in a score between 0 and 100. It reflects an individual's perception of the amount of workload devoted to each of the scales, along with each scale's contribution to overall workload [24]. Here, we analyze the overall workload ratings in addition to the six individual scale ratings. As with typing and error rates, for each analysis a 3 (highlighting) x 2 (phrase set) x 2 (session) ANOVA is used.

5.3.3.1 Overall Workload

An analysis on the overall workload does not reveal any interesting effects. There is no significant main effect for highlighting, phrase set, or session. In addition, there is no significant interaction between highlighting and phrase set, highlighting and session, and phrase set and session. Finally, there is no 3–way interaction between highlighting, phrase set, and session. Although the overall workload score revealed no effects, an analysis of individual workload scales can still reveal relevant information about how the typing task contributes to different sources of workload [20]. For each scale, the rating (0–20) is analyzed without regard to the participant's weighting of that scale. On each scale a higher rating reflects more workload or difficulty.

5.3.3.2 Physical Demand

There is no significant 3–way interaction between highlighting, phrase set, and session. Moreover there is no significant interaction between highlighting and phrase set nor highlighting and session. Finally, there is no significant main effect for highlighting. However, there is a significant interaction between phrase set and session $F_{(1,54)} = 13.72, p < 0.01, MSE = 8.18$. The ordered group rated physical demand lower in the practice session ($M = 8.42, SD = 5.13$) than the evaluation session ($M = 11.27, SD = 5.16$), $F_{(1,54)} = 13.88, p < .01$. The unordered group did not rate physical demand differently between the practice and evaluation session. Simple effects were further examined by analyzing the impact of phrase set in the practice session and the evaluation session. In the practice session the ordered group rated physical demand significantly lower ($M = 8.42, SD = 5.13$) than did the unordered group ($M = 12.63, SD = 5.22$), $F_{(1,54)} = 7.56, p = 0.01, MSE = 29.41$. However, in the evaluation session no significant difference in ratings was found between the two phrase set groups. This suggests that the increase in physical demand between sessions for the ordered group is a result of demand being lowered in the practice session; in the evaluation session the physical demand was not different for either group.

5.3.3.3 Effort

For the effort scale, there is no significant 3–way interaction between highlighting, phrase set, and session. Also, there is no significant interaction between highlighting and phrase set nor between phrase set and session. Furthermore, there is no significant main effect of phrase set indicating that the phrase set manipulation did not change participants' rating of the effort required to type on the Twiddler. The highlighting manipulation does interact with session, $F_{(2,54)} = 8.48, p = 0.001, MSE = 3.86$. A simple effects analysis of session at each level of highlighting reveals that the highlighting off group does not report significantly different amounts of effort between the practice and evaluation sessions. However, the highlighting on group rated the effort required to type in the practice session lower ($M = 13.38, SD = 4.11$) than the effort required in the evaluation session ($M = 14.93, SD = 3.61$), $F_{(1,54)} = 5.64, p = 0.02$. In contrast, the delayed highlighting group reported higher effort in the practice session

($M = 13.80, SD = 3.30$) compared to the evaluation session ($M = 12.70, SD = 3.93$), $F_{(1,54)} = 11.16, p < 0.01$. Further simple effects analyses reveal that the three highlighting groups are not significantly different in either the practice or evaluation session.

5.3.3.4 Mental and Temporal Demand, Performance, and Frustration

The software manipulations do not have any significant effects on mental demand ratings or performance ratings. There is only one significant difference for ratings on the temporal demand scale: a main effect for session. This result indicates that participants rated the evaluation session as more temporally demanding ($M = 10.24, SD = 4.19$) than the practice session ($M = 8.28, SD = 4.09$), $F_{(1,54)} = 12.79, p < 0.01, MSE = 9.07$. Ratings of the frustration scale also yield no significant effects for highlighting, phrase set, or session. It is interesting that there are no effects for session (either a main effect or an interaction with phrase set or highlighting). This result seems to suggest that when the help that was provided in the practice session (such as highlighting on or ordered phrase set) was removed, participants did not feel more discouraged or stressed in the evaluation session.

5.3.4 Comparison to Previous Work

Data from our first study on Twiddler typing rates (Chapter 3) can be used as a baseline against which to compare our current typing rates. Although many differences exist between the two studies which could account for differences in typing rates (e.g., compensation, instructions, error highlighting, phrase set, etc.) we believe the comparison can still be illuminating. In order to compare the two studies we utilized a 2 (session) x 2 (study) ANOVA. The study factor has two levels: previous and current, which correspond to the original study and the current study. This analysis combines the current study's experimental conditions into one group. There is a significant interaction between the session and study factors, $F_{(1,68)} = 27.51, p < 0.01, MSE = 1.19$. A simple effects analysis shows that within the practice session, the current study yielded faster typing rates ($M = 5.99, SD = 1.75$) than the previous study ($M = 4.27, SD = 1.35$), $F_{(1,68)} = 8.84, p < 0.01, MSE = 2.88$. However, within the evaluation session there is no significant difference in typing rates between the current study ($M = 6.15, SD = 1.89$) and the previous study ($M = 7.18, SD = 2.08$),

$F_{(1,68)} = 2.54, p = 0.12, MSE = 3.63$. In the previous study, typing rates increased significantly from the practice session to the evaluation session, $F_{(1,68)} = 35.81, p < 0.01$. However in the current study, typing rates did not significantly change between the two sessions, $F_{(1,68)} = 0.61, p = 0.44$. Together, these results suggest that the current study raised typing rates in the first 20 minutes.

In order to investigate the possiblity that our Twiddler phrase set (as opposed to the MacKenzie phrase set) is responsible for the difference in typing rates for the first condition, we compare our baseline condition (highlighting off and unordered phrase set) to the previous study's data. If there is a difference between baseline conditions we can attribute the change to any of the several differences between the two studies, including the phrase set. We used the same 2 (session) x 2 (study) ANOVA analysis strategy but limited our data set to the baseline condition in our current study and the previous data's study. We therefore used only the data from 10 participants in the new study and the data from the 10 participants in the old study. As before, we found a significant interaction between study and session, $F_{(1,18)} = 5.32, p = 0.03, MSE = 4.87$. A simple effects analysis of study at each level of session shows the practice condition does not have a statistically significant difference between the the old study ($M = 4.27, SD = 1.35$) and the new study ($M = 4.17, SD = 1.74$), $F_{(1,18)} = 1.69, p = 0.21, MSE = 2.41$. Likewise, in the evaluation condition there is no reliable difference between the old study ($M = 7.18, SD = 2.08$) and the new study ($M = 6.69, SD = 1.87$), $F_{(1,18)} = 0.31, p = 0.59, MSE = 3.91$. This result suggests that the phrase set by itself was not enough to alter typing rates across studies.

5.4 Discussion

Across all of our measures, the effects of our two aids are encouraging. In general, using the ordered phrase set and highlighting helps novice Twiddler typists' performance while in use. The ordered phrase set increases typing rates and lowers the subjective physical demand during the practice session. While this effect did not persist in the evaluation session, increasing performance while using the aid may help adoption. Simply presenting the keys to be learned sequentially, in groupings that correspond to the keyboard layout,

allows individuals with no experience to type meaningful phrases faster and with less effort. This result is consistent with existing research that has found training beginners on parts of a task, rather than the whole task is beneficial [11, 28].

Enabling highlighting for the first 20 minutes of typing increases typing rates, reduces the number of errors, and reduces subjective ratings of effort. However, we believe that the results indicate that this highlighting may have a slight cost. Error rates increased and typing rates decreased once highlighting was turned off. While error rates increased, the group with highlighting made no more errors than the groups without; although typing rates decrease, they are not slower than the other groups. Using highlighting with a delay did not have an overall positive effect on typing or error rates. It might be that we did not have the correct timing delay to show any meaningful benefit. The failure to find any significant benefits should not rule out future investigations into the utility of delaying highlighting for novices.

Comparing the data from this study to the first two sessions of our previous Twiddler evaluation shows that our aids are beneficial for the first twenty minutes of typing and do not hinder the second twenty minutes once removed.

We believe these aids would be helpful in convincing prospective users that a Twiddler is easy to adopt even though one types differently than with current mobile phones. For example, in a mobile phone store, a demonstration that featured highlighting might entice potential users to try typing with chords. Once the user bought a mobile phone, a typing tutor on the phone could use the reduced phrase set to provide the user with a quick feeling of accomplishment. Then as the user became more experienced, the MacKenzie phrase set could be used to further the user's skill.

5.5 Conclusions

In this chapter, we presented a study examining two aids designed to help novice typists on the Twiddler mobile one–handed chording keyboard. We found that using an ordered phrase set designed around the Twiddler keymap helps typing rates and reduces physical demand and highlighting reduces error rates and decreases the subjective physical demand.

Given an expert Twiddler user's ability to enter text rapidly in a mobile setting and the ability to help novice typists with our two software aids, chording seems to be a viable mechanism for text entry on future mobile devices.

Our experiments evaluating the Twiddler have shown that we can increase a person's data entry capabilities on mobile devices; the Twiddler offers a rapid touch typing capability and is useable under limited visual feedback conditions. In the next two chapters we shift our focus to our second input theme of dual–purpose speech.

CHAPTER 6

DUAL–PURPOSE SPEECH

In the previous chapters we examined the Twiddler keyboard and found that an expert wearable user can employ this device to rapidly enter information from conversations. Our studies on the Twiddler have shown that novices can be trained to type rapidly after moderate practice. While a very useful input mechanism, the Twiddler does have some drawbacks. Of primary concern is the need to learn to type. As we presented in Chapter 5, our two typing aids can facilitate the novice experience. While these aids are useful, they do not completely eliminate the barrier of learning for novices.

In this chapter and the next, we introduce and evaluate dual–purpose speech: a new mechanism for input that is explicitly designed to be used in conversation and that is more amenable for novice users. A dual–purpose speech interaction is one where the speech serves two roles. First, it is socially appropriate and meaningful in the context of a human–to–human conversation. Second, the speech provides useful input to a computer. A dual–purpose speech application can listen to one side of a conversation to provide beneficial services.

By using speech recognition, we can reduce or eliminate the need to learn how to enter information manually into a mobile device. While this reduces the barrier for novice users, the use of speech for input presents some issues. In a human–to–human conversational situation, it is important that any speech interaction with a computer fits the flow of the conversation. There are numerous situations where it would be socially inappropriate to talk directly to a computer. For example, entering notes verbally into a computer could easily disrupt the flow of a meeting.

With dual–purpose speech we can overcome this problem. Instead of requiring the user to provide extra speech for computer input, we reuse information already in the conversation. A dual–purpose speech application utilizes the content from the user's side of the

conversation and attempts to minimize disruptions in the flow of conversation by reducing manual interaction with the computer. By doing so, the user can maintain speech where the language and grammar used fits the conversation.

We have built three applications targeted at mobile devices which provide explicit support during a conversation that we use to explore the concept of dual–purpose speech: the Calendar Navigator Agent, DialogTabs, and Speech Courier. The Calendar Navigator Agent navigates a user's calendar based on a scheduling dialog with the user's conversational partner. DialogTabs allows a user to postpone cognitive processing of conversational material by providing short–term capture of transient information. Finally, Speech Courier allows asynchronous delivery of relevant conversational information to a third party.

6.1 Calendaring Scenario

Many office workers have adopted the practice of carrying personal digital assistants (PDAs) and other mobile computing technology to assist them during conversations. The computers are used to schedule appointments, take notes, and jot down reminders. Manually entering the information encountered during the conversation is the predominant input mechanism. The following example illustrates the issue. Alice is trying to schedule a meeting with her manager, Bob. *Italics* denote the process of Bob using his PDA:

Alice: Bob, can we meet next week?

Bob pulls out his PDA.

Bob: Next week you said?

Bob starts the scheduling application.

Alice: Yes, how about Monday?

Bob uses his stylus to switch to month view.

Bob: Monday, let me check.

He selects next Monday to change to day view.

I'm busy all day Monday.

Bob advances the calendar one day.

How about Tuesday?

Alice: Tuesday at one then?

Bob selects the 1:00 entry.

Bob: Sounds good. I'll pencil you in at one.

Bob enters Alice's name at 1:00 and puts away his PDA.

This example illustrates some of the current interaction issues with tools used during conversation. Although the information to schedule the appointment was spoken during the conversation, Bob must still manually enter it into his computer. Additionally, it is difficult for Bob to participate in the conversation and navigate through the applications on his PDA at the same time. Instead, he must put Alice "on hold" while he interacts with his scheduler. Past research on mobile calendaring interactions has shown that device access time and time for data entry on mobile devices can lead to disuse [58].

Our proposed technique of utilizing dual–purpose speech allows Bob to converse with Alice while also providing sufficient information for his computer to automatically move through his calendar. Although our method uses speech recognition, our approach does not require Bob to suspend his conversation with Alice. Instead, our application allows Bob's speech to fulfill its traditional conversational role with Alice while also serving as input to his computer. We explore this example more thoroughly in Section 6.4.1.

In the remainder of this chapter we discuss more in–depth the idea of dual–purpose speech and explore the issues involved in mobile speech recognition from the standpoints of technology and privacy. Next, we present the details of three applications which utilize dual–purpose speech. We then discuss the common dual–purpose speech issues raised by these applications and describe the common speech infrastructure we have utilized. In Chapter 7, we present a detailed evaluation of the Calendar Navigator Agent and discuss the ability of novices to use the application.

6.2 Related Work

Our reuse of conversational material with dual–purpose speech centers around the use of speech recognition. The concept of machine speech recognition was popularized by Vannevar Bush in his 1945 Atlantic Monthly article "As We May Think" [7]. A speech interface

was hypothesized to be faster and more "natural" than typing or writing, and initial Wizard of Oz experiments by Gould in 1983 on computer assisted dictation supported this hypothesis [19]. Most public development efforts in the last decade have focused on dictation or interactive voice response systems (e.g. "Show me the flights from Dallas to Pittsburgh") [29]. Despite this early work, speech recognition has not found widespread success, especially with mobile systems. Several researchers have explored the limits of current speech recognition technology and its appropriateness for various interfaces and situations [49, 68, 43, 16, 48]. Shneiderman provides a brief overview of the issues in his "Limits of Speech Recognition" [52], and Cohen and Oviatt provide a detailed list of conditions when speech may be advantageous in "The Role of Voice Input for Human-Machine Communication" [15].

In this work, we employ many speech interface techniques described by these authors to constrain our problem of recognizing speech. Our work is also influenced by systems which forgo speech recognition and store the audio directly, using other cues such as pen strokes, location, or time of day for indexing the audio [61, 60, 64, 67, 22]. Several of these systems are directed at situations when the amount of spoken information is overwhelming, such as attending a conference. By using similar interface techniques, our applications are designed to degrade gracefully despite potential errors with our speech recognition.

Work on human–human communication is also relevant to our use of dual–purpose speech. In particular, Speech Acts Theory states that the act of saying something performs an action [2, 50]. In a dual–purpose setting, one utterance might perform two speech acts: one for the conversational partner and one for the computer. In general it would be difficult to automatically interpret speech acts with a computer because the computer has limited access to the user's history and context, and this information is critical to the meaning of a speech act. Furthermore, people often mean more than what they actually say [50] and the rest of the information is interpreted by the other person using shared context. As we will show, the scope of our applications is sufficiently constrained so that we can make some assumptions about the nature of the speech being used, and all of our applications use push–to–talk so that the user segments the machine relevant portions of the speech.

6.3 Dual–Purpose Speech

Dual–purpose speech may already be familiar to the reader from other settings. For example, a lawyer may have her assistant, Alice, in the office while on the telephone with a colleague. Upon agreeing to exchange some information, she might tell her colleague "My assistant Alice will send you our new proposal today." This utterance is dual purpose; it informs the colleague of the lawyer's intention and provides Alice with the specifics needed to fulfil her instructions without further interaction. We explore this scenario with our Speech Courier application (Section 6.4.3).

We extend the concept of dual–purpose speech to a computer interaction technique. Consider a problem described in 1998 by the Boston Voice Users Group [17]. One of the group members, who used a commercial speech recognition package for his everyday work, noticed that it was inconvenient and socially awkward to disengage the system when guests visited his office. Before he could speak to his guest, he had to turn off the system by saying "Go to sleep." He would then turn to his visitor, say "Just a second" and remove his headset and earpiece. Eventually this individual discovered the solution to his problem. Rather than telling the system "Go to sleep," he changed the stop command for the system to "Just a second." This modification allowed his speech to serve a dual purpose: it disabled the speech recognition system and gracefully informed his guest that he would be ready to converse shortly. The dual–purpose speech transformed a socially awkward situation into one in which a single utterance served two purposes: a command to the computer and a polite comment to the guest.

We have developed this technique as a way to enable computer support during conversations. Effective use of speech as an interaction technique in this domain is challenging. During a human–to–human conversation it is important that any speech interaction with a computer fit the flow of the conversation. There are numerous situations where it would be socially inappropriate to talk directly to a computer. For example the flow of a conversation would be disrupted if a user addressed her computer in the middle of a conversation: "Computer! Show me my schedule for next week." By using dual–purpose speech, a person can maintain socially appropriate speech: speech where the language and grammar used

fits the conversation. While it is important that the language used is socially appropriate it might not be strictly "natural." The user may need to slightly modify her language to effectively use the application. Even so, with dual–purpose speech the resulting conversation still follows social conventions and sounds "natural" to the conversational partner. The applications we present in Section 6.4 utilize the content from the user's side of the conversation and attempt to minimize disruptions in the flow of conversation.

One notable feature of our applications is that they depend only on the speech of one person, the user. Many other projects involving scheduling recognition tasks assume that all sides of the conversation are available [59, 6]. Recording other people's speech without their permission, however, leads to privacy concerns. Also of concern are the limitations of current speech recognition technology.

With only one side of the conversation available, one might think that it is infeasible to obtain all of the required information to complete a task such as scheduling. However, the user can assist the computer by repeating important points that the other person has stated. People often repeat what another person has said to confirm understanding. It is likely that the user already repeats much of the critical information, and the conversational partner is unlikely to realize that the user is repeating any additional information for the benefit of his applications. The example conversation in Section 6.4.1 reflects this behavior.

6.3.1 Privacy

A primary concern with speech recognition is the need to record audio, which can lead to issues with privacy. In most areas of the United States, recording of conversations with electronic devices is permissible if at least one participant in the conversation is aware of the recording. In twelve states, however, all participants in a conversation must give consent for recording in most situations [46].

We have constrained the use of speech in our applications in an effort to preserve privacy. Currently many mobile devices, such as mobile phones, have the ability to record the audio of conversations around the device. Anecdotally, our colleagues have found that it is possible to record people's voices from across a room on some mobile phones. Our primary mechanism

for avoiding this effect and insuring the privacy of others is to use a high quality noise cancelling microphone. Worn near the user's mouth, these microphones cancel out nearly all ambient sounds except for the user's voice. In our experience, this greatly reduces the volume of or eliminates the conversational partner's voice from the captured audio. With this technique, our applications utilize only the user's side of the conversation.

6.3.2 Speech Recognition

Limitations of current speech recognition technology make recognizing meaningful portions of casual conversation very difficult. Mobility significantly confounds speech recognition, resulting in higher error rates and restricting the types of devices and methods that may be used for error correction. Many speech recognizers have not sufficiently addressed the varying noise situations that occur during mobile speech. Bursty street traffic noise and microphone noise due to wind can significantly impact a recognition system through insertion errors.

While recognition systems will continue to improve, some errors must be expected. A key strategy we employ to reduce the number of errors is push–to–talk. With push–to–talk, the user specifies which parts of the conversation the computer should attend to by pressing a button. This greatly simplifies the speech recognition task. Instead of continuously processing speech, the computer only needs to recognize the portions of a conversation marked by the user. These phrases contain higher ratios of known keywords and sentences to out–of–vocabulary words and out–of–grammar sequences. We can further reduce errors by formulating appropriate grammars and vocabularies to be recognized. Phrases are chosen to cue the applications while simultaneously informing the user's conversational partner in a socially acceptable manner. While these restrictions are not ideal, they enable us to explore the uses of dual–purpose speech and might be eased as technology improves.

6.4 Applications

We have developed three applications that utilize dual–purpose speech to assist a user in conversational tasks: the Calendar Navigator Agent, DialogTabs, and Speech Courier. Since many conversations occur while roaming [62], we built our applications so that they can be

used while mobile. These dual–purpose speech applications reduce the amount of manual input and instead reuse material from the conversation.

The Calendar Navigator Agent automatically navigates a user's calendar based on socially appropriate speech used while scheduling appointments. DialogTabs allows a user to postpone cognitive processing of conversational material by providing short–term capture of transient information. Finally, Speech Courier allows a user to alert a non–present third party to relevant material from her conversation.

6.4.1 The Calendar Navigator Agent

The Calendar Navigator Agent (CNA) is a calendar application that has been augmented to utilize the user's speech during a social interaction. The CNA automatically navigates a person's calendar based on a socially appropriate dialog used while creating an appointment. The goal is to allow user interaction with the calendar that has minimal disruption of the scheduling conversation.

When the Calendar Navigator Agent is started, it shows a familiar style of scheduling application (Figure 35a). The graphical interface is similar to common scheduling applications available on PDAs or desktops. As the user proceeds with a conversation, he can hold the "talk" button to run the speech recognition. The speech fragment is processed by the speech recognition engine using a limited grammar tailored to calendaring (for more details, see Section 6.6.1). Specific keywords such as "next week" or "Monday" are recognized by the CNA's speech recognition engine and used to perform specific actions. If an error is made and an improper action is performed, the user can press a single button to undo the last command.

In Section 6.1, we described a motivating scenario for our work in which Alice is trying to schedule an appointment with Bob. We will now revisit that scenario and show how using dual–purpose speech eases the conversation for both participants. Bold face text indicates words spoken by Bob while push–to–talk is active.

The conversation begins with Alice requesting a meeting with Bob.

Alice: Bob, can we meet next week?

Bob starts the CNA (Figure 35a) and presses the "talk" key to activate recording.

Bob: **Next week you said?**

Bob releases the "talk" key to stop recording. The CNA recognizes the key words "next week" in the sentence; knowing the current date, it jumps the display to next week (Figure 35b). As this is occurring, Alice is speaking:

Alice: Yes, how about Monday?

Glancing at Monday on the display, Bob quickly sees several meetings, but it's unclear for how much of the day he'll be occupied.

Bob: **Monday?** Let me check.

The CNA recognizes the keyword "Monday," and switches the view to a close–up of Monday (Figure 35c). It is now clear to Bob that Monday is mostly full. Remembering from the week view that Tuesday seemed clear Bob suggests to Alice:

Bob: I'm busy all day Monday.

How about Tuesday?

The CNA, detecting the keyword "Tuesday," jumps the view to the next day (Figure 35d). Bob can see that he has few appointments. Alice suggests a time.

Alice: Tuesday at one, then?

Bob sees that one o'clock on Tuesday afternoon is free.

Bob: Sounds good. **I'll pencil you in at one.**

The CNA recognizes "one" as a time and creates a new appointment (Figure 35e). Bob may now finish the conversation with Alice. Afterwards, he can fill in the rest of the relevant information for his meeting at his leisure as our speech recognition engine is currently not capable of recognizing the names of people or places.

This conversation is nearly the same as the original; however in this scenario, the amount of information that Bob has to manually enter into the schedule is greatly decreased. Instead, the CNA uses conversational information to navigate the calendar. Bob's interaction with his computer is reduced to using the push–to–talk button, pausing briefly during the conversation to glance at his calendar, and filling in the uncaptured meeting information after the conversation is over.

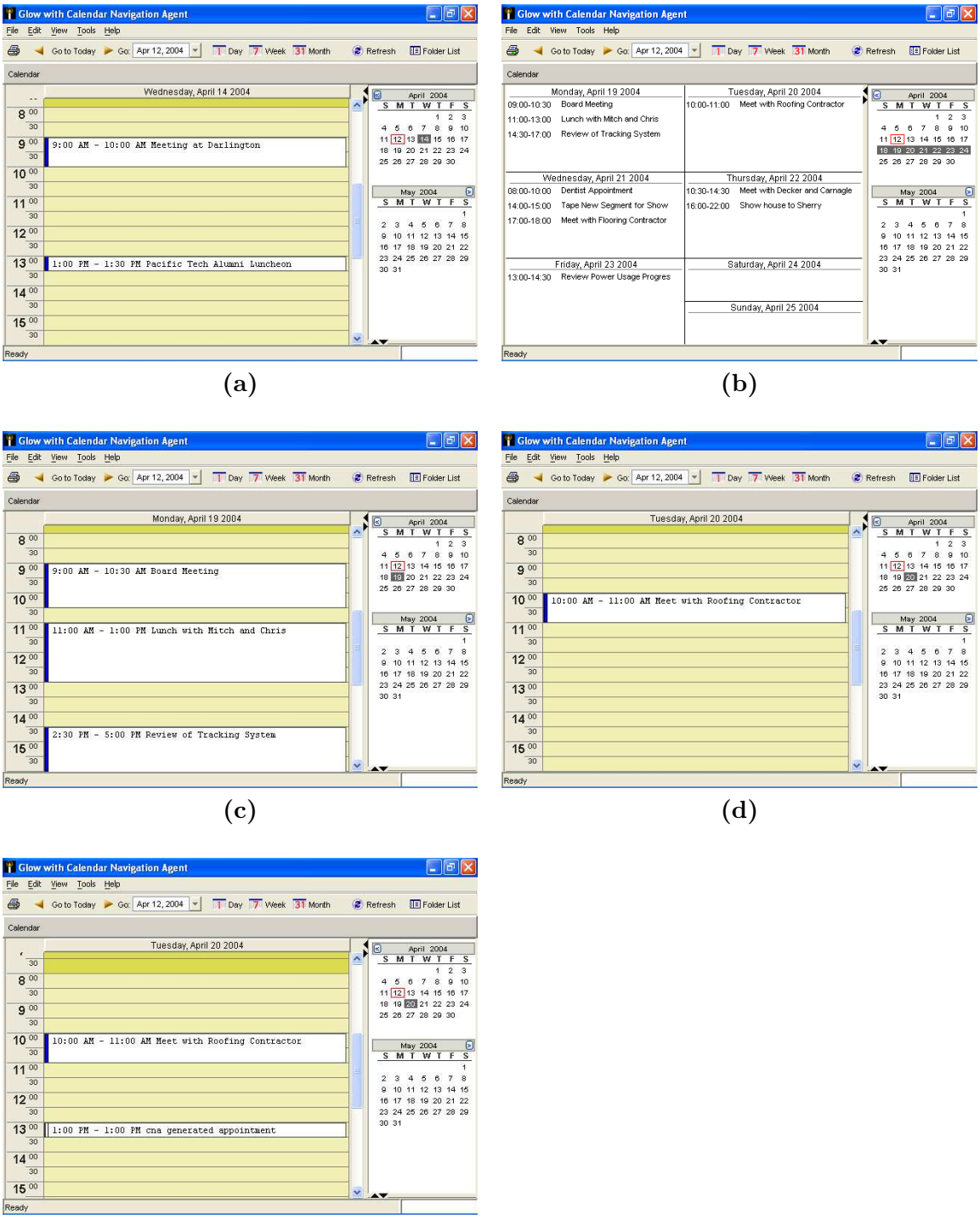

Figure 35: (a) The CNA starts and displays the current date. (b) Cued by "next week," the CNA shows the overview of Bob's schedule the following week. (c) The CNA recognizes "Monday" and shows the detail view for that day. (d) The CNA jumps forward one day when "Tuesday" is recognized. (e) Once the CNA recognizes the time, one o'clock, a new appointment is created.

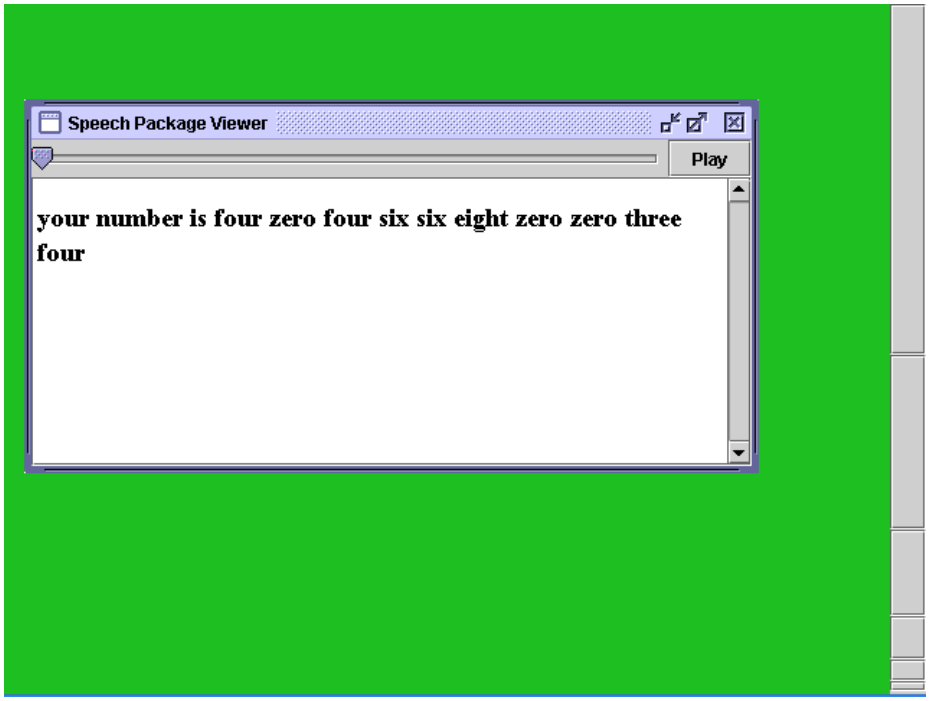

Figure 36: DialogTabs display unobtrusively on the right side of the display. The pop–up allows the user to see the transcribed speech and listen to portions of the audio.

6.4.2 DialogTabs

In the previous example, we show how the CNA allows navigation through a calendar. Bob postponed the job of filling out the details in his scheduler entry until after his conversation was over. A natural extension of the CNA would be to capture the audio for this portion of the conversation. The idea of postponement during a conversation is explicitly supported with our next application, DialogTabs.

DialogTabs is designed to help compensate for the limits of short–term memory. Unlike other short term audio reminders (such as the Personal Audio Loop [22]) DialogTabs only processes the user's side of the conversation and uses a push–to–talk button to segment out the relevant portion of a conversation. A small widget, the Dialog Tab, is created to provide a visual reminder of the recording. After the conversation, the user can re–listen to the postponed audio and view an attempted speech–to–text translation (Figure 36).

Imagine that Bob, after finishing setting up his meeting with Alice, encounters his boss Eve in the hall as she is on her way to an important meeting. Eve has some information

for Bob: she wants him to call a client and quickly tells him the phone number.

Eve: Bob, please call our client about the new proposal.

They are out of the office; the number is 555-1292.

Rather than open the notepad application on his PDA and try to write the number or look for a pen and a scrap of paper, Bob quickly pushes the DialogTabs button on his mobile computer and repeats the number back to Eve.

Bob: **555-1292**. I'll do it now.

When Bob stops recording, DialogTabs creates an unobtrusive tab on the side of Bob's screen; as Bob returns to his phone he can go back and view the tab with the number to make the call. In addition to recording the phone number, Bob exhibits good social etiquette; by repeating the number back to Eve, Bob lets Eve know he heard her correctly.

DialogTabs is explicitly designed to make use of dual–purpose speech. While it could be used as a general short term audio reminder outside of a conversation, using dual–purpose speech makes it well suited as a conversational aid. Many conversations are very short and any time spent diverting attention towards a PDA or paper takes away from the conversation. By reducing the interaction to a single button press and reusing speech from the conversation the cost of the interaction becomes very low.

Visual feedback for each speech segment is generated by showing a Dialog Tab. As they are created, tabs stack vertically in order of arrival. The most recently created tab is the tallest, appearing at the top of the stack and covering twice as much screen space as the next tab. Together the tabs appear as a thin vertical bar at the right edge of the display (Figure 36). During the course of the day, several tabs may queue up, but the user does not need to process them until he has the time and inclination to do so. The stacked tabs provide a reminder of the information that is waiting for attention, so the user can postpone considering the conversational segments without fear of forgetting them. As each tab is created, the system attempts to recognize the segments of speech recorded for each tab. Hovering the mouse over a tab displays the recognized text, while clicking a tab brings up a dialog box showing a visual representation of the recorded audio along with the text (Figure 36). The user can click on words in the text or the scroll widget to hear a segment

of audio.

Creating a grammar for a general purpose DialogTabs application would be very challenging. To address this issue, we have built several different versions of DialogTabs that use task–specific grammars. Our first uses the CNA grammar while another uses a grammar designed to parse phone numbers. However, even in a more general unconstrained case, DialogTabs is designed to be useful with numerous recognition errors. An inaccurate transcript can be sufficient to remind the user of the contents of the conversation fragment, and if not, the user can replay the original audio. Our graphical interface for the transcript is similar to that of the SCANMail system [63], which allows users to visually browse voicemail messages.

6.4.3 Speech Courier

Our final prototype application is Speech Courier. This tool is designed to relay relevant conversational information to an absent third party and was inspired by informal observations of a high level manager and his work routine. Communication and delegation of tasks to the manager's coworkers consumes much of his work day. Several times a day while conversing with a colleague, either face–to–face or on the telephone, a new task for his assistant is generated. Often his assistant is present during the conversation waiting for tasks that might be created.

For example, Eve might say to Bob:

Eve: Yes Bob, Alice will email you the write–up

for our new proposal.

Bob understands he will get an email.

Alice knows to send the email.

Alice is present during the conversation and Eve's speech serves a dual purpose: informing Bob and tasking Alice. Figure 37a depicts this situation. Alice understands the new task even though there was not a separate explicit communication between Eve and Alice. Unfortunately this type of interaction requires Alice to be present for the conversation and limits her ability to do other work. If Alice is not present, Eve needs to remember at some other point to give her the new task. As Eve is very busy and often gets distracted by other

work, she can easily forget to assign the task to Alice. With the manager we observed this happen on several occasions.

Speech Courier can be used to transform the synchronous dual–purpose face–to–face speech of this situation to a remote asynchronous communication. Using Speech Courier, a user can easily capture an important part of the conversation and send it to a non–present third party. The user marks the important points of the conversation using the push–to–talk button as with our other applications. Once the audio is captured, the speech recognition engine generates a transcript and the audio and transcript are bundled into a package and sent to the third party recipient via email. In our implementation, a single "assistant" user is configured to receive the package and they might be used to convey action items, tasks, reminders, or updates to the non–present person.

Returning to our example, if Alice were not present she would not overhear her task. Using Speech Courier, Eve can tag and save the relevant portion of her conversation and send it to Alice. During the conversation with Bob, she uses the Speech Courier button to record the relevant portion of her speech.

Eve: **Yes, Bob, Alice will email you the write–up**

for our new proposal.

Bob understands he will get an email.

Speech Courier sends the task to Alice.

Speech courier creates a package with the audio and a transcript and automatically sends it to Alice (Figure 37b).

Our speech recognition language model for Speech Courier is more broad than CNA or DialogTabs, but still rather limited. The speech recognition produces several errors when words are not in the vocabulary. The recorded speech would likely be sufficient to understand the message but the addition of a mostly correct transcript should improve the utility of the application [63]. Because this information is intended for another person, the user might wish to correct any errors or add additional comments. Speech Courier provides the user the ability to edit the recognized text before sending the package and uses an interface similar to the pop–up of DialogTabs (Figure 35f). As the package is created when

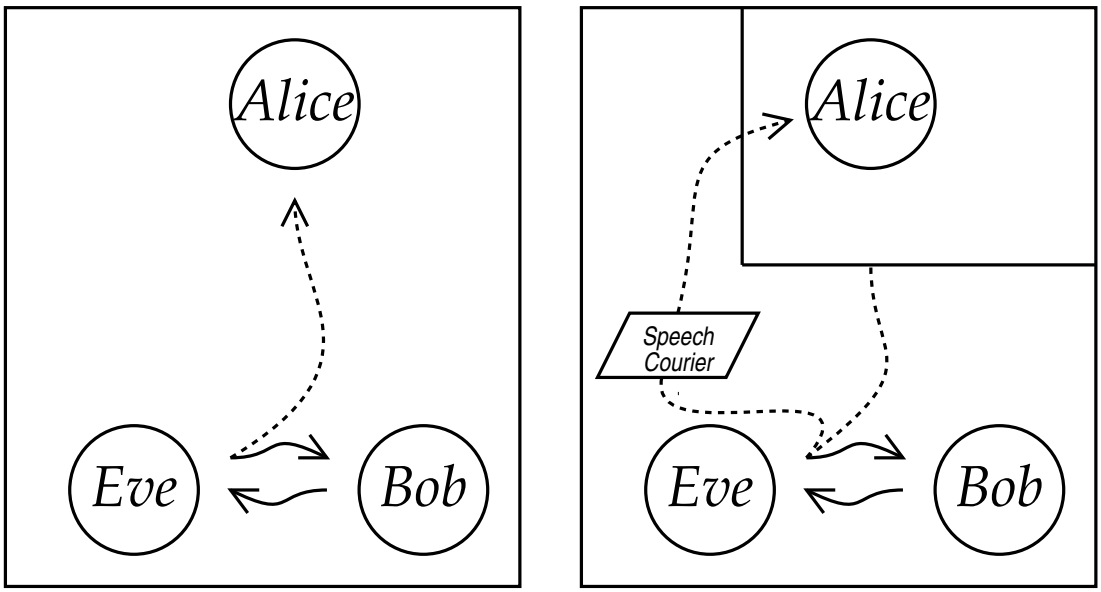

Figure 37: (a) When present, Alice can follow the conversation between Eve and Bob waiting for tasks. (b) When Alice is absent, Eve saves relevant portions of her conversation with Bob using Speech Courier, which then forwards the information to Alice.

the user is engaged in conversation, she would likely delay this interaction until finished. The editing capability allows the transcript to serve as a rough draft for a note destined to the third party.

6.5 Discussion

Calendar Navigator Agent, DialogTabs and Speech Courier all use dual–purpose speech to provide support for a user engaged in conversation. This technique eliminates, postpones, or reduces manual interactions until socially appropriate. While ease and speed of interaction are design considerations in any application, the duration of an interaction is critical when designing tools for use in conversation. Conversations in the office are quite brief; one study shows the average length of a conversation as a mere 113s, while 50% of conversations last less than 38s [62]. Any time spent interacting with a computer can disrupt the flow of conversation. In the worst case, the user will avoid using the tools altogether.

Our dual–purpose speech applications have the advantage of a low cost of failure, where failure is many recognition errors on the part of the speech recognition engine. With DialogTabs and Speech Courier, a completely inaccurate transcript will mean only that the

Table 8: Design matrix of dual–purpose speech. Our applications are restricted and intentional.

	Restricted	Unrestricted
Intentional	CNA, DT, SC	Co–located assistant
Unintentional	N/A	Naive ideal

user must listen to the entire clip of speech which can be done after the conversation is completed. Imperfect recognition on the part of the Calendar Navigator Agent forces the user to address the error. She must either repeat the phrase in a socially appropriate way while avoiding a cascade of errors or revert to manually navigating through the calendar. In the latter case, the rest of the interaction would be the same as if she had used manual input the whole time.

Even though our applications only listen to the user's side of the conversation to protect the privacy of others, they still offer beneficial functionality. In the cases where the dual–purpose speech applications need additional information, the user can repeat back to her conversational partner. The echoing of key dates in the CNA or repeating a phone number during a conversation allows the user to give both input to the computer and also confirm that the message has been heard and understood properly. Repeating key points is often already performed when communication channels are poor or the information is particularly important. For instance, military radio conversations have special protocols to ensure proper communication [1]. Even though repeating back information for use by the computer may mean a small change in communication habits, the privacy benefits of only recording the user's side of the conversation are significant.

6.5.1 Dual–Purpose Speech Design Space

These three applications highlight important aspects of dual–purpose speech. The first issue is if the dual–purpose speech is intentional or unintentional. That is, does the user intentionally speak to both her conversational partner and the computer or just to the conversational partner? Closely related is the question of the language used; it can either be restricted or unrestricted. The next issue concerns the intended recipient of the speech.

The recipient can be a computer or a person, and if it is a computer, it can act upon the speech like the CNA or only passively record and transcribe the speech.

Intentional dual–purpose speech is when the speaker intends for her speech to be directed towards both parties. Unintentional dual–purpose speech is formulated only for a single recipient even though the second is listening and acting. Unrestricted language is natural everyday speech with no boundaries, whereas restricted language requires a predefined limited vocabulary (Table 8). All three of our applications are intentional and use restricted language. At the very least, the user must press the push–to–talk button to segment her speech. She must also intentionally restrict her speech to the language model of the three applications. An example of intentional and unrestricted dual–purpose speech can be found in the scenario that inspired Speech Courier where Eve talks to Bob while Alice is listening (Figure 37a). Eve is explicitly talking to Bob but also formulating her speech so Alice understands. The case of unintentional restricted dual–purpose speech cannot exist because a speaker can only restrict her speech intentionally. Lastly, unintentional unrestricted speech would be the least burdensome for the user, possibly creating a better user experience. This is the goal of many context aware systems such as the Remembrance Agent [47]. Unfortunately, current speech recognition requires some restriction in the user's language to achieve satisfactory results. Furthermore, it is not clear if the user's language would contain enough information to be of use given the implications of Speech Act Theory [2, 50].

The next issue to consider is the intended recipient of the dual–purpose speech information and whether an application acts upon the person's speech or uses it passively. DialogTabs is an example of a passive application that buffers the audio and associated transcript for the user. The recipient of Speech Courier package is a non–present third party. The CNA is an example of an application where the computer is the intended recipient of the speech; the user's speech is mapped directly to actions performed on the calendar.

The intended recipient changes the impact of speech recognition errors. The CNA operates directly on the speech, and an error in speech recognition can result in an improper action with the calendar. If there is an error, the user cannot continue with scheduling until

it is addressed. To correct errors, the user must divert attention away from the conversation, thereby disrupting the flow and distracting the user from her primary task. In contrast, if a person is the recipient such as with Speech Courier or DialogTabs, an error "only" results in an improper transcription. Errors made in applications such as these do not interrupt the conversation and can be corrected during a less demanding time (after the conversation is over). The value of the transcript decreases as the associated errors increase, but the applications still function and remain useful.

6.6 Implementation

The implementation of our dual–purpose speech applications requires a mobile computer such as a wearable or laptop capable of performing speech recognition in near real time, and the user must wear a high quality noise canceling microphone. We also use an input device for push–to–talk and a display for visual feedback. In this section, we discuss how we met these requirements in building the Calendar Navigator Agent, DialogTabs and Speech Courier.

Utilizing a high quality audio source helps to improve the accuracy of speech recognition and ensures that recorded speech is intelligible when played back. To this end, we used a VXI Talk Pro Max microphone which features active noise canceling and voice enhancement. The noise canceling feature filters out nearly all ambient sounds except for the user's voice, while the voice itself is enhanced by limiting distortion caused by breath pops and other sounds at non–speech frequencies.

For automatic speech recognition (ASR), we used version 4 of CMU's Sphinx software [25]. Sphinx 4 is a highly modular, extensible ASR research system written in Java that has an architecture which allows for the use of custom language and acoustic models. Our prototypes consist of the Sphinx recognition engine, libraries that abstract audio, speech recognition, and visualization services, and graphical user interfaces to these services. All system components are written in Java 2 Standard Edition and run under GNU/Linux and Windows XP. Glow[1], an open source Java calendaring application, was modified for use in

[1] http://groupware.openoffice.org/glow/

the CNA application (Section 6.4.1). The applications run on a 1.7GHz Intel Pentium IV Mobile CPU laptop, and previous implementations of the CNA and DialogTabs have run on an 800MHz Transmeta–based wearable computer.

6.6.1 Acoustic and Language Models

A key issue in building applications that utilize speech recognition is the use of acoustic and language models. Acoustic models provide information about the low–level features of speech such as phonemes, while language models provide information about pronunciation and grammar.

In general, acoustic models are separated into speaker dependent and speaker independent models. A speaker dependent model will be more accurate for the particular speaker that provided the acoustic data, while a speaker independent model allows many users to be recognized at the cost of reduced overall performance. Given the high barrier of entry for creating acoustic models, we chose to use the freely available speaker–independent DARPA Resource Management acoustic model[2].

An important part of our research was constructing an appropriate language model to use in dual–speech situations. A language model consists of a pronunciation dictionary and a grammar that specifies how words in that dictionary combine. A grammar can specify that a sequence such as "How about we meet next week" is highly probable while the sequence "How a lot of next meet" is not. When a certain conversational task can be assumed such as appointment scheduling, task specific language can be engineered into the grammar to reduce processing time and to achieve higher recognition accuracy. On the other hand, when no specific task can be assumed, a relaxed grammar must be used which is necessarily less accurate.

In our implementation of DialogTabs, we chose the limited task of saving phone numbers. The corresponding language model represents one extreme along the continuum of grammar constraints. A corpus of eighteen sentences and nineteen words was constructed. The corpus contains variants of the phrase "So your phone number is..." and the digits zero through

[2]http://www.ldc.upenn.edu/Catalog/

nine. The probabilistic language model generated from the corpus contains 19 unigrams, 36 bigrams, and 26 trigrams.

Though still a constrained task, a much more general language model was built for the CNA. The corpus was prepared by observing the language used by participants during a previous study on mobile calendaring [58]. Example phrases include "How about the day after?" and "Let's meet October twelfth." The corpus contains a total of 1007 phrases and the resulting probabilistic language model contains 121 unigrams, 461 bigrams, and 744 trigrams.

We observed that despite the variation in the language used in the calendaring corpus, there is little variation in the intent of the language. We identified three semantically distinct units that could be leveraged for calendar navigation. These are the initial check of a certain date, the subsequent access of other dates when the initial check fails (e.g. the user has a previous engagement), and the final act of confirming the appointment. After recognition is performed on a sentence, keyword matching is applied to determine which of the three actions is intended. For example, finding "March" and finding "20th" would signal the check of "March 20th," even if the spoken sentence was "let's meet in March...how about the 20th?" This keyword–to–intention mapping helps the Calendar Navigator Agent be more flexible in its recognition especially if the user strays outside the language model. This technique in turn reduces the effect of recognition errors and helps to avoid the cost arising from incorrect navigation.

In contrast to the other applications, a more general purpose grammar was used as the starting point for Speech Courier's language model. This choice explores the use of unconstrained speech recognition in conversational situations that are hard to formulate. Our assumption was that any language model we could construct would perform poorly in an arbitrary situation not accounted for by the model. Our approach was to build a base model that could be iteratively extended according to personal experience, informal observations, and future formal usage studies. The base corpus includes one thousand common words as well as the scheduling scenario corpus identified in the CNA language model for a total of 2042 phrases and 1050 words. The probabilistic language model also

contains 2437 bigrams, and 1779 trigrams.

6.7 Evaluation of Speech Recognition

We conducted a preliminary study of the Calendar Navigator Agent to investigate its effectiveness for scheduling appointments. Specifically we are interested in the effectiveness of our speech recognition, the ease of use of push–to–talk, and the users' ability to employ the restricted grammar. We focused on the CNA because speech recognition errors are the most critical in this application of our three and scheduling allows us to explore dual–purpose speech with a straightforward and realistic task for users.

6.7.1 Procedure

Three people from our laboratory used the CNA for this study. Everyone had a passing knowledge of dual–purpose speech and our applications before the study; however, no one had any experience with our prototypes. The trials lasted 60–90 minutes for each person. The CNA ran on a laptop at a desk, and the laptop's screen displayed the application.

The study consists of four parts: a baseline evaluation of speech recognition, a demonstration of the CNA, training in two phases, and finally a test with scheduling appointments using dual–purpose speech. These steps are designed to gradually introduce the users to the language and abilities of the CNA. After the experiment, we administered a questionnaire and conducted an interview.

Using a testing application, we obtain a baseline of each user's speech recognition rates for the language of the CNA. Each user reads through 20 sentences used by the CNA. For each sentence, the subject uses push–to–talk and speaks the presented phrase. If there is an error, they repeat the phrase until it is correct.

The researcher next demonstrates the CNA by navigating the calendar and scheduling two appointments using speech. The user is instructed to listen to the speech and watch the resulting actions taken by the CNA.

Next are the two training phases designed to instruct the users on the association between the speech used for the CNA and the actions performed in the calendar. Users schedule two appointments per training phase in a sequence of steps. Each step represents

one turn of the user's dialog. Part one of the training is the prompted phase. At each step of this phase, the researcher explains the possible actions that can be taken from the current state in the CNA. The researcher then gives the user a phrase to speak, asks her to repeat it to ensure she understands what to say, and the user speaks the phrase to the CNA. The second training phase is the user generated phase. As in the previous phase, the user schedules two appointments step–by–step. However instead of being prompted with what to say, the user is given a more general goal and asked to generate a phrase to use with the CNA. Once the participant generates a correct phrase, she uses it with the CNA.

The last portion of the experiment is the test phase designed to mimic appointment scheduling conversations. The user is asked to participate in nine scheduling dialogs with the researcher. Using the information in the CNA calendar, the user responds to calendaring requests made by the researcher or initiates a dialog given a high–level goal (e.g., "schedule an appointment next week").

At the conclusion of the experiment, we administered a questionnaire composed of 12 Likert scale questions and used the answers as a basis for a semi–structured interview.

6.7.2 Results

While limited in scope, the results from our study are positive. Table 9 shows the word–level recognition rates for our three users taken from our initial speech recognition baseline phase of the experiment. Percent accuracy is defined as $\frac{N-D-S-I}{N} \times 100\%$ and percent correct as $\frac{N-D-S}{N} \times 100\%$ where N is the total number of words, D is the number of deletions, S the number of substitutions and I the number of insertions. Overall, the mean accuracy for the group is 87.0%, while the percent correct is 93.3%. While better recognition rates would improve our application, one user performed very well achieving 100% correctness and 97.5% accuracy on our 20 phrases. This result indicates that with an improved or adapted acoustic model, we might be able to enhance our overall recognition rates.

While the word–level speech recognition rates provide an overall sense of the performance of the application, the actions performed by the CNA are more important. For the testing portion of the study, phrases were successfully recognized and acted upon by the CNA

Table 9: Word level percent accuracy and percent correct for three users.

	A	B	C	Mean
Accuracy	79.9%	97.5%	83.6%	87.0%
Correct	88.9%	100%	91.0%	93.3%

without errors 80.2% of the time. Furthermore, each task in the scheduling dialog was completed with at most one recognition error 97.8% of the time. This result implies that uttering the phrase again seems to be effective for the CNA. Our current language model is very limited and was not designed to enable socially graceful correction of errors through speech. For our experiment, the user was asked to repeat a phrase until the CNA performed the correct action. Given this result, we are exploring ways to modify our language model so that a user can repeat or rephrase what she said. This ability would enable the computer to try again, while at the same time minimizing any disruption in the flow of the conversation. Most of the speech recognition errors in our study resulted in no action taken by the CNA as opposed to the incorrect action. It is possible that using a slightly more intelligent algorithm to interpret the speech might increase the ability of the CNA to perform the correct action when speech recognition errors are made.

The questionnaire and interviews provide additional insight. Our users quickly accommodated to using push–to–talk and rated it as fairly easy to use during the scheduling conversation. Our users thought that the language for scheduling with the CNA was fairly acceptable and socially appropriate. Even with the training given, the users indicated that language generation is the hardest part of using the CNA. This issue was demonstrated most clearly when the user initiates the scheduling dialog and cannot simply respond to the conversational partner. The users also realized their own limitations, and this quote is typical: "This app[lication] would be really useful given more training." Ideally, the use of dual–purpose speech should be much more effortless. Our results imply that the applications should have a better language model so that the user's speech can more closely match her natural language. While difficult in general, a similar effect could be achieved with an effort similar to the DARPA Airline Travel Information Service (ATIS) task where researchers try to capture the "natural" vocabulary and grammar related to a specific task and then create

a system that allows very flexible natural interaction while still being specifically tuned to the task [23, 29, 31]. Even with the current limitations, all three users were enthusiastic about the CNA application and agreed using conversations and dual–purpose speech as a means to schedule appointments would be useful. In Chapter 7 we present a more detailed evaluation of the CNA.

6.8 Conclusions

We introduced the concept of a dual–purpose speech interaction: socially appropriate speech that provides meaningful input to a computer. We showed that dual–purpose speech can be employed by applications to augment conversations. Our three applications, the Calendar Navigator Agent, DialogTabs, and Speech Courier, explored this design space, and we identified three aspects of dual–purpose speech: restricted language, intentional use of speech, and intended recipient. We discussed issues of designing interactions based only on the user's speech to ensure privacy and robustness in the presence of speech recognition errors. With future improvements to speech recognition, we expect dual–purpose speech to become more widely applicable for mobile computing.

CHAPTER 7

EVALUATION OF DUAL–PURPOSE SPEECH

The results from our initial study in the previous Chapter on dual–purpose speech are encouraging. We found that users could perform calendaring operations with the Calendar Navigator Agent with modest training even with the restrictions on language imposed by our prototype's speech recognition engine. In this chapter, we present our final experiment which evaluates dual–purpose speech more thoroughly. The experiment is designed to determine whether or not novice participants can remain engaged in a dialog while at the same time use their speech to control the computer. Furthermore, we are interested in the strategies developed while using dual–purpose speech. For instance, what types of dual–purpose speech utterances are constructed, and how are they woven into the conversation?

7.1 Calendaring Interaction

To evaluate these issues, our study is constructed so novice dual–purpose speech users schedule a sequence of appointments with a researcher using the Calendar Navigator Agent on a personal digital assistant (PDA). For each appointment, the researcher initiates a calendaring discussion with a goal, e.g. "Can we meet next week?" The participant then navigates the calendar on the PDA we provide and negotiates a suitable time with the researcher. After a time is agreed upon, an appointment is created. For this experiment, we are evaluating dual–purpose speech, as well as collecting a baseline of traditional pen input.

With this data, we hope to answer the following questions. Can novices successfully operate the calendar while engaged in a conversation? What strategies do our participants adopt in using dual–purpose speech? What are the costs of using dual–purpose speech relative to more traditional mobile input techniques? In particular, are there differences for performance, cognitive load, or any difference on the impact of the input technique on the

scheduling conversation as a whole?

7.1.1 Calendaring Dialog

In a real world scenario, one of the two conversational partners initiates a scheduling dialog with a goal. Unfortunately, this situation is not practical in our experimental setting. Instead, the researcher initiates all of the appointments for this study. While not entirely realistic, it is a more reasonable alternative than requiring the participant to create or be prompted with the artificial scheduling tasks. Adding this burden would be a confounding factor in the use of the scheduling device.

Another change we impose on the calendaring interaction for the experiment is the exclusion of the details associated with a new appointment. Normally, once a suitable time is negotiated, the people engaged in the conversation might also determine who else will be at the meeting, the meeting's location, etc. This information is typically stored in the calendar entry. For this study, we are not addressing the issue of entering these types of details associated with an appointment. Instead we are only evaluating the performance for navigating the calendar and initial appointment creation. Once a suitable time for both parties has been found, a default appointment of one hour is created. While the details surrounding an appointment are important, this modification is useful. While removing this portion of the appointment procedure limits external validity, the alternative has similar problems. As with the above issue of the participant initiating a calendaring dialog, the participant would need to negotiate artificial details for the appointments created. This situation is also unrealistic given the artificial nature of the task which lacks the context of a real appointment.

7.1.2 Wizard of Oz

We are utilizing the Wizard of Oz technique in place of an automatic speech recognition system. A second researcher, known as the wizard, simulates the recognition and semantic processing of the participants' speech. This technique has a long tradition in the evaluation of speech systems and was used by Gould in his early studies on the use of speech for computer input [19]. By simulating the recognition of speech, researchers can focus on the

evaluation of speech in the interface before exerting the large engineering effort needed to build an effective and accurate speech recognition system.

In addition, this technique enables flexibility in the language we accept with our dual–purpose speech applications. To build a real speech recognition system, one needs to create an accurate acoustic model of the user's speech and environment as well as a language model representative of what the user can say. Our prototypes from the previous chapter use a freely available acoustic model with limited coverage (Section 6.6.1). In particular, the model is designed for dictation and not conversational speech. For our prototypes, we constructed a very limited language model based upon our usage scenarios. As dual–purpose speech is a new technique, we do not yet know exactly what language users may want to employ, and therefore constructing an effective language model would be problematic. We can overcome these technical issues by using a wizard.

Finally, using a wizard allows us to test part of the dual–purpose speech design space that is not currently practical. As discussed in Section 6.5.1, there are several options available when creating a dual–purpose speech interaction. For example, the original CNA prototype uses push–to–talk and restricted speech to facilitate speech recognition. For this study with novice users, we wanted to ease the language requirement. In contrast to our original system which has a very constrained language model, we are using unrestricted speech for input. By using the Wizard of Oz technique, we can allow the participants to use their preexisting scheduling language with the system. We also retained the push–to–talk functionality from our previous implementation to enforce intentional dual–purpose speech: the user must intentionally depress the button before speaking to the calendar. This act reminds the participants to use a phrase that can be understood by the system as well as their conversational partner (the researcher). It also helps filter out extra speech that might cause errant or unexpected commands.

7.2 Design

For our experiment, we have a dual–purpose speech condition and a control condition using pen input. Both conditions are performed by every participant resulting in a within

Table 10: Examples of appointments scheduled during experiment.

Can we meet...
on a Wednesday morning?
February 17th?
last Monday of March?
Friday?
tomorrow morning?
The week of March 21st?
on a Monday afternoon?

subjects design, and the order of conditions are counterbalanced across participants. For the pen input method, the participant uses the PDA stylus to navigate the calendaring application. In the speech condition, the participant uses dual–purpose speech. During each condition, the participant performs twenty appointment creation trials with the given input method. Each trial is untimed, and the entire experiment takes approximately 45 minutes per participant.

7.2.1 Trials

The researcher scheduled twenty unique appointments with the participant for each condition. He initiated all of the scheduling dialogs with a phrase similar to "Can we meet..." That phrase was then followed by one of 40 different predefined scheduling goals. Table 10 shows some examples of the appointments.

7.2.2 Participants

We recruited twenty participants from the Institute. All of the participants were compensated $10 for their time. Our participants ranged from 19 to 39 years old and had a mean age of 25.9 years ($SD = 5.8$). Fifteen participants were male. Nineteen of our twenty participants reported that they owned a cell phone, while six indicated they owned a PDA. Our participants reported that they spent an average of 14.4 hours per week in scheduled meetings or classes ($SD = 7.0$). Fourteen of the participants indicated they had prior experience with speech recognition, many with automated phone response systems.

7.2.3 Procedure

The experiment began with the researcher presenting an overview of the experiment, and consent and demographic forms were filled out. Next, the participants were given written instructions describing the calendar application they would be using on the PDA. The instructions detailed the different views of the calendar (day, week, month) and described the different interface elements. The participants were informed that they would be making a series of appointments with the researcher and to use the schedule on the PDA as their own. They were also told that all of the appointments would be made for 1 hour and start on the hour. If there were no questions, the participant put on our headset microphone used to capture speech and the trials for the first condition started.

For the pen condition, the participant navigated the calendar using the PDA stylus for input. The session began with some practice interactions designed to familiarize the participant with the pen input mechanism. The researcher instructed the participant to navigate through a sequence of days and weeks, and after completing the pre-defined navigations and appointment creations, the participant was instructed to try a few interactions of his or her choosing. Once the participant was satisfied, the trials began.

For the dual–purpose speech condition, the participant navigated the calendar using speech input. Again, the session began with practice. The researcher informed the participant how to use the push–to–talk button and asked him to perform a simple navigation using speech. Next, the researcher described dual–purpose speech and instructed the participant that he could use a single utterance to fill two roles. The researcher then stepped the participant through a simple dialog that showed how to use dual–purpose speech and how the speech affected the calendar. After the predefined navigations were complete, the participant used speech input to control the calendar at his own discretion. Once satisfied, the practice ended and trials began.

The calendar software was preloaded with a schedule that the participant used as his own during the experiment. For the trials, the researcher prompted the participant with an appointment, and all twenty appointments were completed in sequence. As the participant proceeded through the trials, appointments accumulated in the calendar. At the end

of each condition, the participant completed a NASA Task Load Index and Likert scale questionnaire described below.

After the trials for the first input method were completed, the participant took a five minute break. After the break, the calendar was reset, the participants put the headset back on, and the study resumed using the other input method. The condition started again with practice and then moved on to a different set of twenty appointments. After the second set of trials was finished, the experiment was complete.

7.2.4 Software and Equipment

The experiment was conducted in the usability laboratory as a stationary test. The participant and researcher sat at a table facing each other. The participant was provided with an iPaq PDA and our calendaring software. The wizard was located in an adjacent room out of sight.

A computer next to the researcher ran the software for the experiment. The core of the software was a modified version of the calendar from the GPE Palmtop Environment[1]. GPE is a collection of open source software which runs on Linux and uses the X Window System. GPE provides a suite of personal information management (PIM) tools and is often used as the software on Linux based PDAs. Figures 38, 39 and 40 show the calendar presented to the user.

The GPE calendar was modified in several ways for this experiment. First, we made a few adjustments to the graphical user interface. The only major modification visible to the participant was the removal of a popup dialog box that is used to enter additional information about an appointment. Normally, this popup is displayed when a user selects a date. Since we are not entering any additional information for the appointments in this experiment, we modified the behavior so that selecting a date creates a default appointment with a duration of one hour (the duration of all of the appointments to be created in the study). We also modified the software so that one of the buttons on the front of the PDA would act as an undo. The undo functionality worked for both pen and speech conditions

[1] http://gpe.handhelds.org/

Figure 38: The day view of the calendar.

and when pressed reversed the last action performed by the user. The final user interface modification added the push–to–talk functionality for speech input. Again, we used one of the buttons on the front of the PDA for this feature.

As discussed above, we did not use machine speech recognition system and instead simulated one with a wizard. We implemented this functionality by routing the captured audio of the participant's voice to the wizard in the adjacent room. In normal operation, the wizard's audio is muted, and he cannot hear any of the appointment dialog. In the speech condition, the push–to–talk button unmutes the wizard's audio. As a result, the wizard hears only the portions of the participant's speech when the button is depressed, which in turn enables him to simulate a speech recognition system.

The wizard interface is actually an extension of the GPE calendar (Figure 41). The calendar was modified with additional windows to allow quick navigation and appointment creation. In addition to the traditional view of the PDA calendar (Figure 41 top, left), the wizard also had access to a window that allowed him to navigate between the days of two months (Figure 41 middle, left). He used this interface to directly navigate between the different views of the calendar based on the participant's speech. He also had another window which he used in the creation of appointments (Figure 41 middle, right). To create

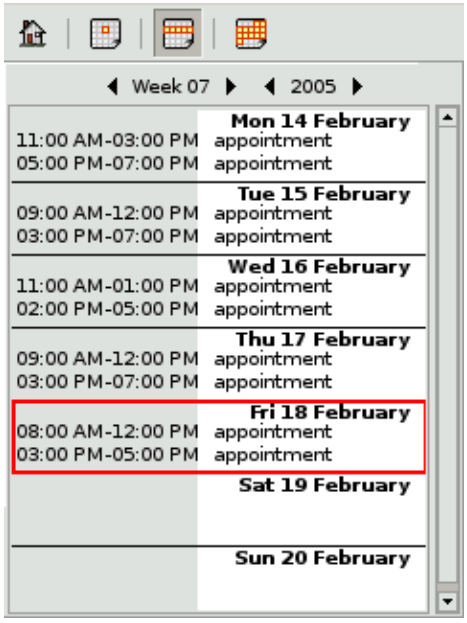

Figure 39: The week view of the calendar.

an appointment, the wizard clicked on the time the participant said. The xterm shown at the bottom of the figure was used by the wizard to start and stop the experimental software.

The calendar had one additional user interface component (Figure 41, top right) which was used by the researcher in the room with the participant. This component was used to control the flow of the experiment. It started and stopped the logging for each individual trial and also prompted the researcher with the appointment identification number to use for any given trial. Finally, the researcher and wizard used a text editor (not shown) as a shared text channel to communicate as needed.

The software ran on a computer sitting next to the researcher. The iPaq PDA was connected to the computer through USB, and the wizard's computer connected to it over the wired network. The one calendar program was then shared with the iPaq and wizard through VNC[2]. VNC is an application that can export the graphical user interface of a program to remote computers and allow remote users to interact with that program. For this experiment, we used VNC to export only the main calendar window to the PDA so that the participant saw what appeared to be a traditional PDA calendar. In reality, it

[2]http://www.realvnc.com/

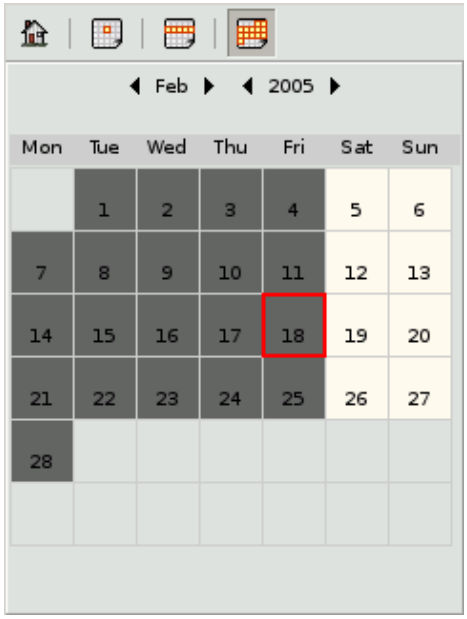

Figure 40: The month view of the calendar.

is actually just a window into the complete experimental setup (Figure 41 top, left) which appeared on the PDA similar to Figure 38. The wizard received a full copy of the interface which he used to drive the application during the speech condition, and the researcher used the application to step through the different conditions and trials.

There were also several modifications made to the software for the purposes of data collection. First, the calendar was augmented so that any change in the calendar's state was recorded to a log. For instance, we logged whenever the calendar switched between the different views (day, week, month). We also logged whenever an appointment was created (and under which condition). Finally, we recorded the use of the buttons on the PDA. In particular, we logged when the push–to–talk button was pressed and released, and when the undo button was used.

The software also recorded audio from two microphones. The first microphone was the headset worn by the participant. It is a VXI Talk Pro Max microphone and only recorded the participant's speech. The audio from this microphone was routed to the wizard and muted and unmuted as described above. The second microphone was placed on the desk and recorded the entire conversation.

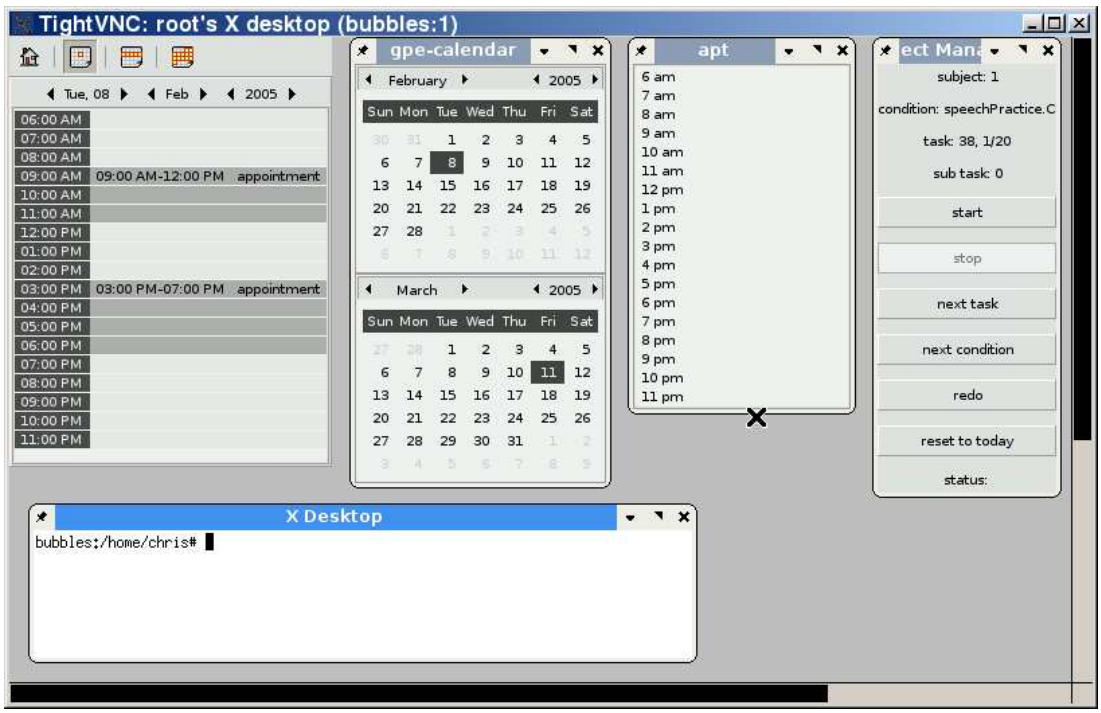

Figure 41: The wizard interface used to quickly schedule appointments and control the flow of the experiment.

7.2.5 Dependent Variables

There were three primary sources of data in this experiment. First, we recorded audio from the conversation of the user interacting with the PDA and researcher. Second, as described above, our software automatically logged several variables such as the duration of each trial and the navigations performed within the calendar. Finally, we collected subjective data on the user's experience with questionnaires.

The audio data was manually coded by the researcher for the participants' turns, the portion of the conversation where a participant held the floor. The audio from the headset microphone was used in the visualization program shown in Figure 42. This program shows the wave form of the audio and has the ability to display data from the log file and overlay it on the waveform. For instance, we can visualize when the push–to–talk button was held down, when the undo button was employed, when each navigation occurred, and when appointments were made. While coding the audio, the annotations were turned off, and the data was coded blind with respect to the input method used.

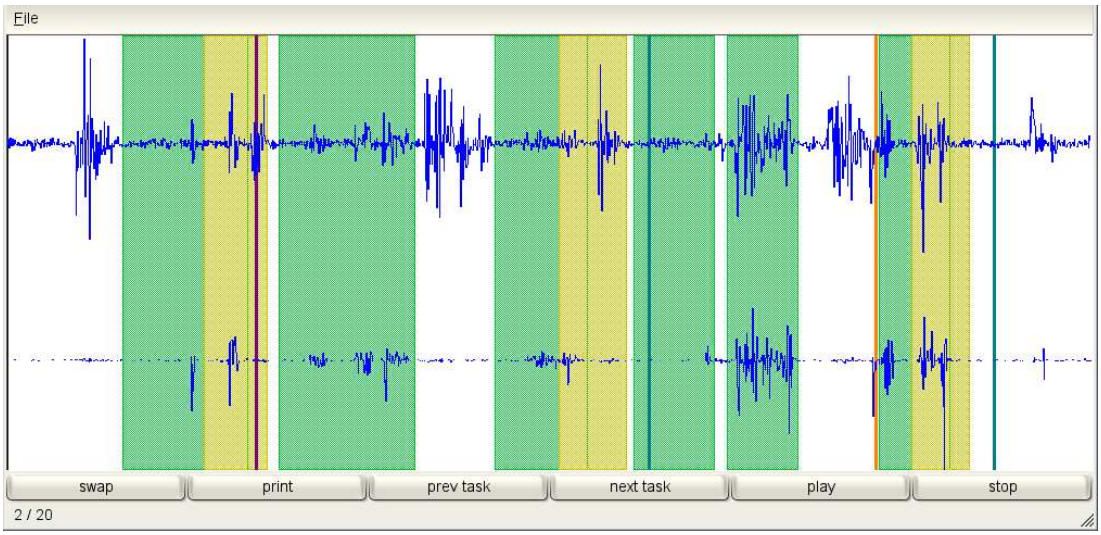

Figure 42: The visualization program used to display and annotate the audio and event data collected. This screen–shot shows an annotated dual–purpose speech trial.

Figure 42 shows the data from a dual–purpose speech trial. The top waveform shows the ambient audio that was recorded, and the bottom waveform corresponds to the audio recorded by the headset microphone worn by the participant. Color coded vertical bars represent different events. The purple line above the 'p' in 'print' corresponds to a navigation. The green line above the 'a' in 'next task' and 's' in 'stop are appointments, and the orange line to the right of 'play' shows the use of the undo button. There are also shaded regions of two different colors. The smaller yellow regions indicate when the push–to–talk button was held, and the large green regions indicates the region coded for the participant's turn. Figure 43 shows a pen trial with several navigations, an appointment, and the coded turns.

In coding the data for the participants' turns, we were looking for when they were holding the floor. A turn started when the researcher finished speaking and ended when the participant finished speaking. Occasionally there would be brief interruptions or confirmations during the turn. If the speech added new information to the dialog it counted towards a turn break (e.g. when the researcher said "yes" in response to a question). If there was not any additional information, it did not break the turn. The speech of this form was mostly single syllable utterances, and they were often used to keep the conversation flowing and convey that the person was listening.

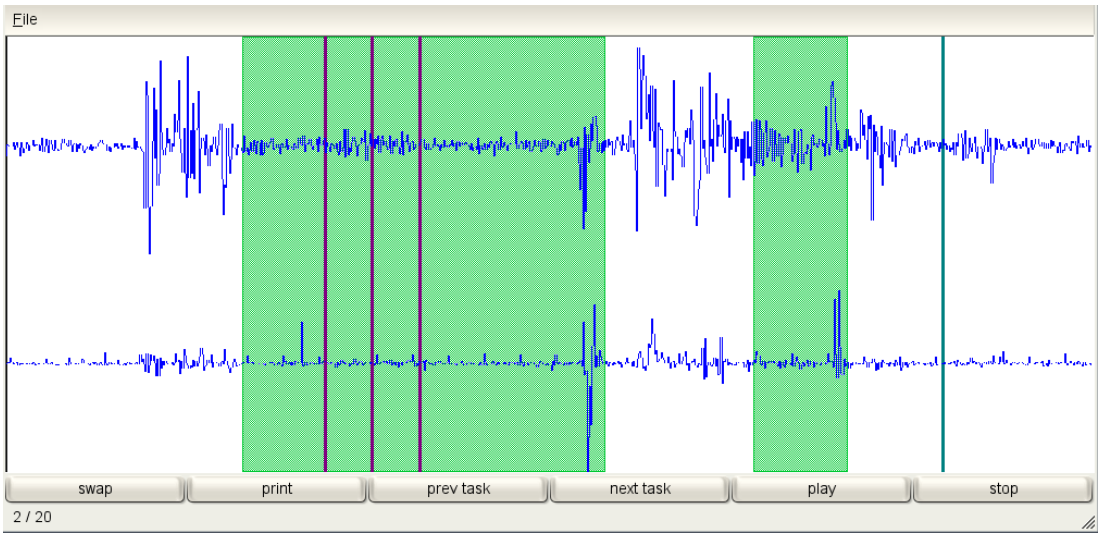

Figure 43: The visualization program used to display and annotate the audio and event data collected.

Several of our other dependent variables were extracted automatically using the coding for turns and the event log of calendar use. The total time for the conversation was calculated as the time from when the participant first spoke (the beginning of her first turn) to when the last appointment was created or the end of the last turn, whichever was later. By starting at the participant's first turn we eliminate the variability in describing the appointment goal to the participant. The end time is not as straightforward. Sometimes the user would finish speaking and then create an appointment (either with the pen or as a result from the delay in the processing of speech by the wizard). At other times, the appointment would be made while still speaking during the final turn. By taking the later of these events we get the total duration of the conversation.

Our next metric was the duration of the user's turn and number of turns during a trial. This information is taken directly from the coded audio data. The cumulative duration of turns is the sum of the duration of each turn in any given trial and represents the amount of time the participant held the floor during a trial.

The number of navigations metric represents the number of interactions the participant had with the PDA that changed the calendar state. For instance, any time the participant changed from one view to another, a navigation is logged.

Finally, we have two metrics that examine when the navigations happen relative to a turn in the conversation. Our analysis software classified each navigation using the coding from the audio data as happening during the turn (while the participant was holding the floor) or while the researcher was speaking (between participants' turns).

7.2.5.1 Questionnaires

At the end of each condition the participants filled out a NASA Task Load Index (TLX) questionnaire [20] (similar to the study in Chapter 5). The TLX provides a measure of the workload imposed by our task and gives a subjective rating on the possible differences between the use of pen and speech input.

At the end of each condition, we also used a Likert scale questionnaire to collect data on the quality of the interaction and conversation. We are particularly interested in any disruption the use of the PDA and input technique may have on the conversation, and this questionnaire is designed to uncover these issues. The following are the five questions we asked:

1. How much did the use of the PDA interfere with the scheduling conversation?
2. How easy was it to carry out the conversation using this input method?
3. Do you feel that using the PDA with this input method disrupted the flow of the conversation?
4. How natural was your speech during the conversation?
5. Did the use of the PDA cause awkward pauses in the conversation?

7.3 Findings

For each of our twenty participants we collected twenty appointment dialogs using speech for input and twenty more using pen. In total we have 800 different calendaring conversations.

7.3.1 Comparing Pen and Speech Input

7.3.1.1 Quantitative Results

We are interested in the relative performance differences between our two input conditions. In general, we found small differences in our metrics which are summarized in Table 11. The conversations were slightly shorter for the pen input condition, taking an average of approximately 3 seconds less time ($M_p = 20.888s, M_s = 17.849s$). In contrast, excluding the researcher's side of the conversation and examining only the duration of each individual turn during the dialog yields no significant difference $p = 0.385$. Comparing the cumulative duration for all of the participants' turns during a task shows the conversation is 2.2s shorter when using pen ($M_p = 13.167$) relative to speech ($M_s = 15.386$). Likewise, comparing the number of turns per trial shows that there were slightly more turns taken using speech ($M_s = 2.875$) than with pen ($M_p = 2.542$). In summary, the participants held the floor slightly longer and more frequently when using speech for input, and the overall conversation took a few seconds longer. Together this data implies that the participants were speaking more in the speech condition.

One of the strongest differences between the conditions is the number of navigations used for an appointment. Here we are using the term navigation to denote any change in the state of the application. For instance, the following all count as navigations: switching from day view to week view, advancing a week, or selecting a particular day. With speech, our participants performed $M_s = 1.3$ navigations per appointment dialog. In contrast, the pen users performed on average an extra 2 navigations during each conversation ($M_p = 3.331$). While speech is not necessarily faster, it has the advantage of being more direct. The speech user can navigate to her intended location in the calendar in fewer steps.

We are also interested in the possibility that the two input methods might offer different affordances with respect to the user's ability to parallel process information. While we do not have data for this trait in general, we can look at our participants' ability to parallelize their input while speaking and listening. Dual–purpose speech is designed to provide parallel input while the user is speaking. During the user's turn in the conversation, she can at her own discretion also input data into the computer by crafting her speech and using the

Table 11: Quantitative results from pen and speech conditions.

Metric	Speech	Pen	P-value
Duration of conversation	20.888s	17.849s	< 0.001
Duration of turn	5.352s	5.179s	0.385
Cumulative duration of turns	15.386s	13.167s	< 0.001
Number of turns	2.875	2.542	< 0.001
Number of navigations	1.301	3.331	< 0.001
Navigations during turns	1.295	3.235	< 0.001
Navigations between turns	0.185	0.210	0.458

push–to–talk button. However, she must wait until it is her turn to talk. It would be very socially awkward to try and enter information using speech input while the other person is speaking. In contrast, the pen user does not have that constraint. She has the opportunity to enter data while their conversational partner is speaking.

To test our participants' use of parallelizing input, we analyzed the number of navigations made while they were holding the floor (during their turns) relative to the number of navigations made while the researcher was speaking (between their turns). First we look at the navigations during the participants' turns. The overall trend observed above in the number of navigations holds here as well. Participants using speech are much more direct in navigating the calendar. It is interesting to note, that nearly all of the navigations occur during the user's turn for both speech and pen input. There is not a statistically significant difference between the number of navigations that happen while the researcher was speaking. Furthermore, the number of navigations that occur during that portion of the conversation is also extremely low with $M_s = 0.185$ navigations for speech and $M_p = 0.210$ for pen. In general it might be possible to parallelize pen input while listening to someone speak. However, this data shows that this happened very rarely for this particular task.

Finally, we were interested in the use of the push–to–talk button and characterizing the delay of speech processing from the wizard. On average, our participants held the push–to–talk button for 1.309s ($SD = 0.67$). Our wizard took an average of 1.513s ($SD = 0.86$) to complete an action once the participant released the push–to–talk button. Changing the delay in speech processing could help increase the performance of using speech for input. As the speed of mobile computers increases the time needed to process the user's speech

Table 12: NASA–TLX results for pen and speech conditions. Statistically significant results marked with *.

Metric	Speech	Pen	P-value
Mental Demand	27.60	27.20	0.915
Physical Demand	2.35	10.10	0.034*
Temporal Demand	22.55	22.05	0.930
Performance	13.10	5.55	0.028*
Effort	24.05	21.90	0.657
Frustration	15.45	10.80	0.439

similarly decreases. With a fast enough computer, eventually this delay could be negligible.

7.3.1.2 NASA Task Load Index

The results from the NASA-TLX are presented in Table 12. This table shows the weighted scores which range from 0, indicating no workload, to 100, indicating very high workload. There is no effect for overall workload, and likewise most of the subcomponents also show no effect between our conditions. The two dimensions that do have a significant difference are physical demand and performance.

Participants rated the speech input method to be less physically demanding than the pen input method ($M_s = 2.35$ and $M_p = 10.10$ respectively). This result is not surprising, given that the only manual input required for the speech condition is to operate the push–to–talk and undo buttons.

In contrast, participants rated their performance to be better using pen compared to speech ($M_p = 5.55$ and $M_s = 13.10$ respectively). There are several possible explanations for this difference. First, none of the participants had used speech recognition in this way before, and several participants mentioned after the session that they were a bit hesitant in their use of speech. This reservation is possibly a result of past experience with the errors that are common in speech recognition systems. While our participants could be quite creative with their speech since we were using the Wizard of Oz technique, the participants were not aware of this possibility and several indicated that they held back not knowing the limitations of the system. If this is the case, the difference in the performance rating would likely diminish as users gained more experience with the system.

Table 13: Questionnaire results for pen and speech conditions.

Metric	Speech	Pen	P-value
1. Interference from PDA?	7.800	6.100	0.238
2. Ease of use of input?	5.550	6.300	0.585
3. Disrupt flow of conversation?	7.300	8.050	0.662
4. Naturalness of speech?	9.000	6.850	0.189
5. Awkward pauses?	8.800	8.600	0.893

7.3.1.3 Input Questionnaire

Our final data point is the Likert scale questionnaire we administered to gain insight on the naturalness and flow of the calendaring conversations (listed in Section 7.2.5.1). As we discussed in Chapter 6, one has to be careful when using speech for computer input in social situations. While dual–purpose speech is designed to reuse conversational material and thus fit the flow of the conversation, we wanted to test this design principle in practice.

Table 13 shows the results from five Likert scale questions we asked at the end of each condition. For this questionnaire we used the same scale as the NASA–TLX for consistency. The minimum value is 0 and indicates a positive experience, while 20 is the maximum and implies that there was a disruption in the conversation. There is no significant difference between the speech and pen conditions for any of our five questions. This result implies that our participants did not think that one input method or the other caused more disruption to the flow of the conversation.

7.3.2 Use of Dual–Purpose Speech

In the previous section we detailed the relative performance of pen and speech input. Now we focus on some of the more qualitative aspects uncovered by our study.

First is the nature of the speech used. The idea behind dual–purpose speech is that the user can create an utterance that very naturally fits into the flow of the conversation with her partner, but at the same time, provides input to the computer. Our participants had varying degrees of success in using speech as we intended. At one extreme, one participant used very structured speech that was directed primarily at the computer. For instance, to navigate the calendar he would use a phrase such as “Show me the 23rd” “Show me the

following week" "Create an appointment at 2pm on Tuesday the 5th." After the experiment, the participant indicated that he was not sure what the speech recognition system could understand. Therefore, he intentionally restricted what he would say and decided not to vary his speech during the conversation in fear of the system not understanding him. It is possible that this participant's prior knowledge of the limitations of speech recognition led to this behavior.

In contrast, the rest of our participants were much more fluid with their use of speech during most of the conversations. For instance, they might say "Let me check the 23rd" or "Would Wednesday work?" While the participants were often explicitly addressing the computer, the speech also fit into the context of the conversation. The primary exception (besides the one participant described above) was at the beginning of the conversation. As described in our experimental design, the researcher initiated all of the scheduling dialogs. As a result, the participant would often echo the exact same phrase so that the computer could act upon it. For instance, the researcher would say: "Can we meet on February 17th" and the participant would echo "February 17th." Occasionally participants would change their intonation and echo the phrase as a question seeking confirmation from the researcher that they heard correctly, but given the large number of trials each participant performed, they often did not persist with this strategy.

In a real calendaring scenario this behavior would likely not be an issue. First, many of the appointments would be initiated by the CNA user and the initial utterance could be used for input. But even for the cases where the user does not initiate, the strategy of echoing for confirmation could be quite useful. While not effective for our experiment where we scheduled 20 appointments in succession, it might be practical if the appointments were spread over a larger period of time such as over the course of a day or week. It is also interesting to note that several of our participants would do this confirmation echo during the pen condition even before using the speech for input. Often they would speak more quietly or to themselves while they were using the pen to navigate, but the strategy already exists for these participants.

Finally, some of our participants became very adept at weaving the phrases needed for

input into the conversation which are very representative of what we intended for dual–purpose speech. In particular, several participants independently developed a strategy that we have named "speculative scheduling" which involves a creative use of the undo button. Originally, we only intended the undo to be used as a way to deal with errors in the speech recognition which still occasionally happened even with the wizard.

At the end of a dialog when a time is decided, the conversation might proceed as follows. The participant would suggest "How about 2 o'clock?" and the researcher would respond, "2 works for me." In the speech condition some participants would then just echo "2" again while pressing the push–to–talk button to enter the appointment. Other participants, however, used speculative scheduling. They would use the push–to–talk button during the suggestion of "How about 2 o'clock?" If the researcher agreed, the appointment was already completed, and the dialog was done. If the time was not good, the participant would then press the undo button erasing the appointment and either create a new one using the researcher's suggestion or suggest a new appointment themselves. By using the undo functionality in this way, they could create an appointment using truly dual–purpose speech. If they were successful the task would be finished, and if they were not, the cost of removing the appointment was extremely low (a single button press). This strategy is particularly interesting not only because it represents a good example of dual–purpose speech, but also because several participants discovered it independently without any instruction to do so.

7.4 Conclusions

Our data show that novices can effectively use the Calendar Navigator Agent in a conversation while using speech for input. Our speech condition did not show a performance benefit but instead resulted in a conversation where the participant held the conversational floor longer. Speech input is also much more direct than pen; our participants needed fewer navigations during the scheduling dialog when using speech.

Our subjective data on possible disruptions the PDA and input method may have on the conversation showed no statistically significant differences between the methods, and likewise for most of the NASA–TLX results. The TLX did show that our participants

thought they performed worse using speech, but that result could be due to the subjects' inexperience with the use of speech input in this novel way. Finally, the participants rated speech as less physically demanding than pen.

While dual–purpose speech is explicitly designed to be used while the user is speaking, it is possible that our participants could have used the pen input during the researcher's turn of the conversation, much like how our expert enters data with the Twiddler in Chapter 2. Interestingly we did not find this behavior. Instead, for both conditions, the vast majority of the navigations that occurred within the calendar happened during the user's turn. It is possible that the potential time savings of parallelizing input is not perceived as a large issue during a calendaring conversation.

One of the most interesting and unexpected uses of dual–purpose speech was for "speculative scheduling" where the participant used dual–purpose speech to create an appointment and used the undo if it did not fit the needs of his partner. While not all of our participants discovered this strategy, our novices successfully adopted dual–purpose speech. Even with very little training, our participants quickly determined how to construct their speech so the researcher understood it and the computer could act upon it.

Together, the data from this experiment show that users can effectively use dual–purpose speech for input. Novice users quickly accommodate to the technique, and it offers an additional input option which can be employed while a user is engaged in a conversation.

CHAPTER 8

FUTURE WORK AND CONCLUSIONS

8.1 Future Work

There are numerous areas of future exploration that were uncovered while performing this research, and there are other mobile input issues that can be explored. We are interested in evolving our dual–purpose speech applications and deploying them on mobile devices to examine how they are used in everyday life. There is also a large amount of work that would be needed to bridge the gap between the functionality we simulated using a wizard in our detailed evaluation of the CNA and what currently exists in our prototypes that use automatic speech recognition. The corpus of data collected from our evaluation is an important first step in constructing a language model of speech for the CNA.

For the Twiddler, we are interested in exploring more familiar designs that incorporate similar chording capabilities. We are exploring a mobile phone based on the current Twiddler keyboard (Figure 44). For messaging or learning to type, a high resolution screen could be leveraged for a tutorial with our typing aids from Chapter 5. With the rapid ability to type, this device might enable advanced mobile phone features such as mobile email and open the door for other new applications.

Our research on the Twiddler has also spawned work looking at other mobile keyboards

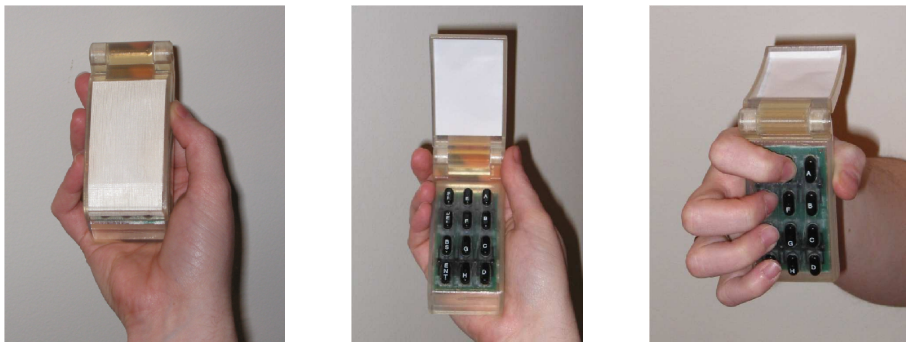

Figure 44: A mobile phone design which incorporates chording capabilities.

such as the mini–QWERTY keyboard common on devices such as Research in Motion's Blackberry [14]. It would also be very interesting to examine in more detail the reasons behind why the Twiddler works so well. One step in that direction would be to create an analytical model of Twiddler typing.

As is necessary with any dissertation, this work focused on a narrow slice of the issues involved with supporting everyday activities with mobile computing. Similar detailed research should also be conducted on other fundamental human–computer interaction issues for mobile computing. For instance, the wearable computing community has adopted head–mounted displays. What are the relative advantages of those displays versus handheld displays or even personal projection displays? What are the right interaction techniques for mobile computing? Exploring the basic input and output capabilities is important, but how are those combined and how should they be used to provided interactive capabilities? And finally, what everyday tasks should be supported? We investigated how to improve support of conversations, but there are many other everyday tasks that could likely benefit from mobile technology.

8.2 Conclusions

Our hypothesis presented in Chapter 1 is that we can enhance mobile input during conversation via two complementary methods:

- By increasing a user's data entry capability with the Twiddler chording keyboard, and
- Through reusing conversational information with dual–purpose speech

We have demonstrated our claims in this dissertation as follows:

- Chapter 1 introduced the need to examine the ability to enter data while engaged in conversations.
- Chapter 2 provides a case study examining practices of an expert wearable computer user. In particular, it highlighted the potential of using computational support during conversations and motivated our detailed evaluation of the Twiddler.

- Chapter 3 presented our initial longitudinal evaluation of the Twiddler chording keyboard. Previous work showed that mobile phone data entry rates are quite limited. In contrast, our participants rapidly learned to type on the Twiddler and at rates faster than other input techniques that utilize a similar keypad.

- Chapter 4 continued our analysis of Twiddler learning rates and our data showed that after approximately 25 hours of practice, previously novice users could type at a rate of 47 wpm. Once trained, we evaluated our participants' ability to enter text with limited visual feedback. We found that they could enter information in our "blind" conditions, and in some cases it actually improved their entry speeds and accuracy.

- While experts can type very rapidly on the Twiddler, Chapter 5 shifted focus back to novice users. We showed that our two different typing aids, highlighting an on–screen keyboard representation and a structured phrase set, increased novice performance and subjective measures of workload while in use.

- Chapter 6 presented dual–purpose speech, a new technique which leverages speech for input that is explicitly designed to be used in conversation and that is more amenable for novice users than the Twiddler. We presented our three prototypes the Calendar Navigator Agent, DialogTabs, and Speech Courier and discussed their design and use.

- Chapter 7 showed that dual–purpose speech can be effectively used by novice users. The data from our final experiment demonstrated that novices could use the CNA to navigate a calendar on a PDA while participating in a scheduling dialog.

This work shows that we can increase a user's data entry potential with the Twiddler chording keyboard, and that novices can reuse conversational material with dual–purpose speech. Furthermore, these two complementary techniques explored in this dissertation represent two different points in the mobile input design space which are useful in a conversational setting. The Twiddler is particularly advantageous to expert users. The ability of the user to touch type and enter information while listening to others speak is also a key feature. In contrast, our data shows novices can rapidly incorporate dual–purpose speech

into a conversation which enables users to reuse information for computer input while they speak. Taken as a whole we believe that this dissertation provides evidence supporting our hypothesis and that we can improve the support of conversations by enhancing mobile computer input.

REFERENCES

[1] Allied Communications Publication, *Communication Instructions Radiotelephone Procedures*, September 2001.

[2] Austin, J. L., *How to do Things with Words*. Harvard University Press, 1962.

[3] Baker, S., GreenWith, H., Einhorn, B., Ihlwan, M., Reinhardt, A., Greene, J., and Edwards, C., "Big bang!." BusinessWeek, June 2004. http://www.businessweek.com/magazine/content/04_25/b3888601.htm.

[4] Beyer, H. and Holtzblat, K., *Contextual Design: Defining Customer-Centered Systems*, ch. Contextual Inquiry in Practice. Morgan Kaufmann, 1998.

[5] Brodie, J., "Designing to support communication on the move," in *CHI '03 extended abstracts on Human factors in computing systems*, pp. 908–909, ACM Press, 2003.

[6] Busemann, S., Declerck, T., Diagne, A. K., Dini, L., Klein, J., and Schmeier, S., "Natural language dialogue service for appointment scheduling agents," Tech. Rep. RR-97-02, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, 1997.

[7] Bush, V., "As we may think," *Atlantic Monthly*, vol. 76, pp. 101–108, July 1945.

[8] Butts, L. and Cockburn, A., "An evaluation of mobile phone text input methods," in *Proceedings of the Australsian User Interfaces Conference*, 2002.

[9] Campbell, C. and Maglio, P., "Supporting notable information in office work," in *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pp. 902–903, ACM Press, 2003.

[10] Card, S., Moran, T. P., and Newell, A., *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum, 1983.

[11] Catrambone, R. and Carroll, J. M., "Learning a word processing system with training wheels and guided exploration," in *CHI '87: Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface*, pp. 169–174, ACM Press, 1987.

[12] CellularOnline, "Women embracing SMS - study." http://www.mobileoffice.co.za/news_2002/100202-women_embracing_sms.htm, October 2002.

[13] CellularOnline, "Stats snapshot and analysis." http://www.cellular.co.za, 2003.

[14] Clarkson, E., Clawson, J., Lyons, K., and Starner, T., "An empirical study of typing rates on mini–qwerty keyboards," in *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pp. 1288–1291, ACM Press, 2005.

[15] Cohen, P. and Oviatt, S., "The role of voice input for human-machine communication," in *Proceedings of the National Academy of Sciences*, vol. 92, pp. 9921–9927, 1995.

[16] Danis, C., Comerford, L., Janke, E., Davies, K., Vries, J. D., and Bertrand, A., "Storywriter: a speech oriented editor," in *CHI '94: Conference companion on Human factors in computing systems*, pp. 277–278, ACM Press, 1994.

[17] DelPapa, J., "Personal communication," *Boston Voice Users Group*, June 1998.

[18] Frohlich, D., "Requirements for interpersonal information management," in *Personal Information Systems: Business Applications*, Stanley Thornes in association with Unicorn Seminars, 1995.

[19] Gould, J., Conti, J., and Hovanyecz, T., "Composing letters with a simulated listening typewriter," *Communications of the ACM*, vol. 26, pp. 295–308, April 1983.

[20] Hart, S. G. and Staveland, L. E., *Human Mental Workload*, ch. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. North-Holland, 1988.

[21] Hayes, G., Pierce, J. S., and Abowd, G. D., "Practices for capturing short important thoughts," in *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pp. 904–905, ACM Press, 2003.

[22] Hayes, G. R., Patel, S. N., Truong, K. N., Iachello, G., Kientz, J. A., Farmer, R., and Abowd, G. D., "The personal audio loop: Designing a ubiquitous audio-based memory aid," in *Proceedings of Mobile HCI*, 2004.

[23] Hemphill, C. T., Godfrey, J. J., and Doddington, G. R., "The ATIS spoken language systems pilot corpus," in *Proc. of the Speech and Natural Language Workshop*, (Hidden Valley, PA), pp. 96–101, 1990.

[24] Hill, S. G., Iavecchia, H. P., Byers, J. C., Bittner, A. C., Zaklad, A. L., and Christ, R. E., "Comparison of four subjective workload rating scales," *Human Factors*, vol. 34, pp. 429–439, August 1992.

[25] Huang, X., Alleva, F., wuen Hon, H., andKai Fu Lee, M.-Y. H., and Rosenfeld, R., "The Sphinx-II speech recognition system: An overview," *Computer, Speech and Language*, pp. 137–148, 1993.

[26] James, C. L. and Reischel, K. M., "Text input for mobile devices: comparing model prediction to actual performance," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 365–371, ACM Press, 2001.

[27] Kidd, A., "The marks are on the knowledge worker," in *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 186–191, ACM Press, 1994.

[28] Kirlik, A., Fisk, A. D., Walker, N., and Rothrock, L., *Making decisions under stress: Implications for individual and team training*, ch. Feedback Augmentation and Part-Task Practice in Training Dynamic Decision-Making Skills. American Psychological Association, 1998.

[29] Kubala, F., Anastasakos, A., Makhoul, J., Nguyen, L., Schwartz, R., and Zavaliagkos, G., "Comparative experiments on large vocabulary speech recognition," in *ICASSP*, (Adelaide, Australia), 1994.

[30] Leech, G., Rayson, P., and Wilson, A., *Word Frequencies in Written and Spoken English: based on the British National Corpus*. Longman, 2001.

[31] Levin, E., Pieraccini, R., and Eckert, W., "A stochastic model of human-machine interaction for learning dialog strategies," *Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 11–23, 2000.

[32] Lin, M., Lutters, W. G., and Kim, T. S., "Understanding the micronote lifecycle: improving mobile support for informal note taking," in *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 687–694, ACM Press, 2004.

[33] Lindstom, M., "Message madness our big chance." SMH http://www.smh.com.au, February 2002.

[34] Lyons, K. and Starner, T., "Mobile capture for wearable computer usability testing," in *Proceedings of IEEE International Symposium on Wearable Computing (ISWC 2001)*, (Zurich, Switerland), 2001.

[35] Lyons, K., "Everyday wearable computer use: A case study of an expert user," in *Proceedings of Mobile HCI*, pp. 61–75, 2003.

[36] Lyons, K., Starner, T., Plaisted, D., Fusia, J., Lyons, A., Drew, A., and Looney, E., "Twiddler typing: One-handed chording text entry for mobile phones," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press, 2004.

[37] MacKenzie, I. S., "KSPC (keystrokes per character) as a characteristic of text entry techniques," in *Proceedings of Mobile HCI 2002*, pp. 195–210, 2002.

[38] MacKenzie, I. S., Kober, H., Smith, D., Jones, T., and Skepner, E., "Letterwise: prefix-based disambiguation for mobile text input," in *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pp. 111–120, ACM Press, 2001.

[39] MacKenzie, I. S. and Soukoreff, R. W., "Phrase sets for evaluating text entry techniques," in *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pp. 754–755, ACM Press, 2003.

[40] MacKenzie, I. S. and Zhang, S. X., "The design and evaluation of a high-performance soft keyboard," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 25–31, ACM Press, 1999.

[41] Matias, E., MacKenzie, I. S., and Buxton, W., "One-handed touch typing on a qwerty keyboard," *Human-Computer Interaction*, 1996.

[42] Mobile CommerceNet http://www.mobile.seitti.com, January 2002.

[43] Oviatt, S., "Ten myths of multimodal interaction," *Communications of the ACM*, vol. 42, no. 11, pp. 74–81, 1999.

[44] Panko, R., "Managerial communication patterns," *Journal of Organisational Computing*, 1992.

[45] Perry, M., O'Hara, K., Sellen, A., Brown, B., and Harper, R., "Dealing with mobility: understanding access anytime, anywhere," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 8, no. 4, pp. 323–347, 2001.

[46] The Reporters Committee for Freedom of the Press, 1815 N. Fort Myer Drive, Suite 900, Arlington, VA 22209, *Can We Tape?*, 2003. http://www.rcfp.org/taping/.

[47] Rhodes, B. J., *Just-In-Time Information Retrieval*. PhD thesis, MIT Media Laboratory, Cambridge, MA, May 2000.

[48] Rudnicky, A., "Mode preference in a simple data-retrieval task," in *ARPA Human Language Technology Workshop*, (Princeton, New Jersey), March 1993.

[49] Schmandt, C., *Voice Communication with Computers*. New York: Van Nostrand Reinhold, 1994.

[50] Searle, J. R., *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, 1969.

[51] Shaffer, L. H., "Multiple attention in continuous verbal tasks," in *Attention and Performance V* (P, M. R. and Dornic, S., eds.), 1975.

[52] Shneiderman, B., "The limits of speech recognition," *Communications of the ACM*, vol. 43, September 2000.

[53] Silfverberg, M., "Using mobile keypads with limited visual feedback: Implications to handheld and wearable devices," in *Proceedings of Mobile HCI 2003*, pp. 76–90, 2003.

[54] Silfverberg, M., MacKenzie, I. S., and Korhonen, P., "Predicting text entry speed on mobile phones," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 9–16, ACM Press, 2000.

[55] Soukoreff, R. W., "Text entry for mobile systems: Models, measures, and analyses for text entry research," Master's thesis, York University, 2002.

[56] Soukoreff, R. W. and MacKenzie, I. S., "Metrics for text entry research: an evaluation of msd and kspc, and a new unified error metric," in *Proceedings of the conference on Human factors in computing systems*, pp. 113–120, ACM Press, 2003.

[57] Starner, T., *Wearable Computing and Context Awareness*. PhD thesis, MIT Media Laboratory, Cambridge, MA, May 1999.

[58] Starner, T. E., Snoeck, C. M., Wong, B. A., , and McGuire, R. M., "Use of mobile appointment scheduling devices," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press, 2004.

[59] Stede, M., Haas, S., and Küssner, U., "Tracking and understanding temporal descriptions in dialogue.," Verbmobil-Report 232, Technische Universität Berlin, October 1998.

[60] Stifelman, L. J., "Augmenting real-world objects: a paper-based audio notebook," in *CHI '96: Conference companion on Human factors in computing systems*, pp. 199–200, ACM Press, 1996.

[61] Stifelman, L. J., Arons, B., Schmandt, C., and Hulteen, E. A., "Voicenotes: a speech interface for a hand-held voice notetaker," in *CHI '93: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 179–186, ACM Press, 1993.

[62] Whittaker, S., Frohlich, D., and Daly-Jones, O., "Informal workplace communication: what is it like and how might we support it?," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 131–137, ACM Press, 1994.

[63] Whittaker, S., Hirschberg, J., Amento, B., Stark, L., Bacchiani, M., Isenhour, P., Stead, L., Zamchick, G., and Rosenberg, A., "Scanmail: a voicemail interface that makes speech browsable, readable and searchable," in *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 275–282, ACM Press, 2002.

[64] Whittaker, S., Hyland, P., and Wiley, M., "Filochat: handwritten notes provide access to recorded conversations," in *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 271–277, ACM Press, 1994.

[65] Wigdor, D. and Balakrishnan, R., "TiltText: using tilt for text input to mobile phones," in *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pp. 81–90, ACM Press, 2003.

[66] Wigdor, D. and Balakrishnan, R., "A comparison of consecutive and concurrent input text entry techniques for mobile phones," in *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 81–88, ACM Press, 2004.

[67] Wilcox, L. D., Schilit, B. N., and Sawhney, N., "Dynomite: a dynamically organized ink and audio notebook," in *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 186–193, ACM Press, 1997.

[68] Yankelovich, N., Levow, G.-A., and Marx, M., "Designing speechacts: issues in speech user interfaces," in *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 369–376, ACM Press/Addison-Wesley Publishing Co., 1995.

[69] Yeh, Y. and Wickens, C. D., "Dissociation of performance and subjective measures of workload," *Human Factors*, vol. 30, pp. 111–120, February 1988.